GMU/SAL-08-502

AFOSR FINAL PERFORMANCE REPORT
FEB 2005 TO MARCH 2008

Principal Investigator: Abbas K. Zaidi
Co-Principal Investigator: Alexander H. Levis

Institution: George Mason University

Full Address: **System Architectures Lab**
ECE Dept. , MS 1G5
George Mason University
Fairfax, VA 22030

Phone: **(703) 993 1774**
Fax: **(703) 993 1601**
Email: **szaidi2@gmu.edu**

Project Title: **Interactive Planning for Capability Driven Air & Space Operations**

Contract/Grant Number: F49550-05-1-0106

System Architectures Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 04/30/2008	**2. REPORT TYPE** Final Performance Report	**3. DATES COVERED** *(From - To)* 02/01/2005-03/31/2008

4. TITLE AND SUBTITLE **Interactive Planning for Capability Driven Air & Space Operations**	**5a. CONTRACT NUMBER** FA9550-05-1-0106 **5b. GRANT NUMBER** (GMU) 200832 **5c. PROGRAM ELEMENT NUMBER** N/A
6. AUTHOR(S) Abbas K. Zaidi	**5d. PROJECT NUMBER** N/A **5e. TASK NUMBER** N/A **5f. WORK UNIT NUMBER** N/A
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) System Architectures Laboratory George Mason University Fairfax, VA 22030	**8. PERFORMING ORGANIZATION REPORT NUMBER** GMU/SAL-08-502
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM 875 North Randolph Street Suite 325 Arlington, VA 22203	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFOSR/NM **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The report describes the progress made, during the reporting period (March 01, 2005 to Aug. 31, 2006), on research conducted to develop approaches for capability driven planning, and to identify/develop methodologies and tools to implement the approach. The report presents work on a temporal representational and reasoning formalism and its software implementation. It also presents findings on an examination of the need and nature of campaign of experimentation to explore approaches to planning in the context of network centric operations.

15. SUBJECT TERMS

Capability driven planning, Temporal logic, Point-interval logic, TEMPER, Campaign of experimentation, Network centric environment

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT **Unclassified**	**b. ABSTRACT** **Unclassified**	**c. THIS PAGE** **Unclassified**	UU		Abbas K. Zaidi **19b. TELEPHONE NUMBER** *(include area code)* 703-993-1774

Table of Contents

1. STATEMENT OF WORK

The following tasks were identified in the grant proposal with some minor terminology changes, mentioned in the productivity report for the year 2005.

Task 1: Extend PIL for Interactive Programming

The present implementation of Point-Interval Logic (PIL) is equipped with a verification mechanism. A fast revision algorithm incorporates the change in an existing system of PIL statements by first identifying the extent of change and then applying the change to the effected part of the system only. In this task, we will integrate these two mechanisms. The two mechanisms, collaborating with each other, are expected to yield more efficient algorithms for handling change in the plans. Further research will be conducted to exploit the advantages of the two approaches to facilitate a dynamic, collaborative, and interactive environment in which several users can input their specifications, in a real-time manner, based on their incomplete picture of the domain at hand.

Task 2: Integrate Space-Time using PIL

The PIL formalism has been shown to incorporate temporal and spatial information separately from each other. The objective of this task is to integrate the two formalisms, temporal and spatial, in such a manner that a causal relationship between temporal and spatial objects can also be captured by the approach, and supported by a fast revision capability. The insistence on fast revision capability is kept to avoid restarting the planning process with every change in the temporal and spatial parameters.

Task 3: Extend the Capabilities of PIL for Space-Time-Capability Integration

The capability-driven interactive planning is presented as a constraint satisfaction problem, with constraints representing temporal, spatial, and capability attributes' requirements for the mission under consideration. The objective of this task is to capitalize on the temporal and spatial constraint-handling capabilities of the PIL formalism for possible use in a first-order algebra, or an applications suite, that captures all three aspects, with an arbitrary degree of abstraction, within a single framework. The analysis tools developed for the present implementation of PIL inference engine will be extended to validate the input specifications and to generate robust feasible plans satisfying all spatial, temporal, and capability-attribute constraints. The graph-theoretic Point Graph (PG) approach used in the knowledge representation and reasoning of spatiotemporal systems, will be extended to incorporate capability attribute features and the causal relationship among all the three aspects of information.

Task 4: Develop Tools for Capability-Driven Interactive Planning

This task consolidates the results of the earlier tasks. The objective is to implement the formalisms for capability-driven interactive planning in a suite of software applications.

Task 5: Planning and Execution in a Network-Centric Environment

This task involves an examination of the need for and the nature of a campaign of experimentation, and an associated program of research, to explore approaches to planning in the context of network centric operations in support of complex, coalition, civil-military missions.

2. STATUS OF EFFORT

Task 1

The inference and revision algorithms of PIL have been updated and incorporated in the application TEMPER. The current inference mechanism has been shown to be complete for a class of temporal queries. The two algorithms, inference and revision, have been combined in a bounding function based solution approach to solving the general temporal problem. The solution uses a combination of the inference and revision algorithms together with a heuristic to implement a fast search in the solution space. The worst-case complexity of the algorithm is exponential as is to be expected for any exact algorithm for an NP Complete problem. An empirical study of the performance of general temporal problem solver using different heuristics has been conducted and the results are presented in a recent conference paper (see Section 9.3).

The scheduling algorithms of TEMPER have also been extended to incorporate stretch float, lead/lag times, milestones for start and finish of activities, and to solve the management problem at the graph representation without a need for the mathematical program required by the earlier approach. The proofs of correctness for the scheduling algorithms are also provided. A new representation of Point Graphs (PG) called Hierarchical PGs has been introduced and implemented in the application that supports both *top-down* and *bottom-up* approaches to project management/planning. The final implementation of TEMPER supports both approaches with some manual steps required to carry out the two. A detailed description of the features, mentioned above, is provided in the two journal papers in Section 9.1 and Section 9.2. A technical report based on the unaltered MS thesis of Mr. Mashhood Ishaque, submitted to the Volgenau School of Information Technology and Engineering in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, has been submitted with an earlier progress report. The thesis was submitted in May 2006.

The PIL formalism has been shown to have applications in mission planning, project management, temporal assessment of situational influence models, and criminal forensics. Some of these applications are presented in attached papers in Section 9.4 and 9.5.

Task 2

A framework for integrating the space and time information into a single formalism was developed. The framework uses the WebTAS[1] suite of application as the underlying platform for storing and visualizing temporal and spatial information. It employs temporal and spatial reasoning tools, together with WebTAS, to store and reason with both qualitative and quantitative information. The inference made by the spatiotemporal tools is fed to the WebTAS database where it can be visualized together with the existing information using WebTAS' extensive suite of interfac-

[1] http://www.issinc.com/webtas/webtas_overview.php

es. The framework is still under investigation and a prototype application is being implemented for an expected release in fall 2008.

Task 3

An attempt has been made to analytically formulate the problem of capability driven planning by identifying the type and nature of constraints/knowledge representation required to handle capability requirements together with temporal and spatial constraints. Some exploratory work was conducted in the use of *ontologies* for mapping capabilities to requirements.

On a parallel track, an attempt is made to develop analytical models that relate strategic objectives/goals, and components of an environment (i.e., political, military, economic, social, infrastructure, and information), to tactical actions for planning and assessment of Effects Based Operations (EBO). A major accomplishment in this regard was a comprehensive expansion and improvement of an existing effects based analytic approach, called Timed Influence Net (TIN) modeling, which allows the adoption of task-sensitive and time-varying flexible influence relationships and functions to derive consistently probabilities of occurrence of sequential events. The theory also provides the means for the utilization of data pertinent to one variable towards the assessment of another dependent variable, and encompasses specific algorithmic steps for the representations and effects of time delays. A variant of this approach, called Activation Timed Influence Networks (ATIN), was developed to capture the explicit mechanisms and/or tactical actions responsible for change in the state of an environment as a result of occurrence of events. An ATIN represents a progressively evolving sequence of actions, where the effects of an action become the preconditions of the action that follows. An ATIN integrates the notions of time and uncertainty in a network model, where nodes explicitly represent mechanisms and/or tactical actions that are responsible for changes in the state of a domain.

The theoretical developments were followed by experimental design and implementation of the algorithms for both enhanced TIN and ATIN models in the application suite called *Pythia*. Sections 9.6 and 9.7 provide detailed technical accounts in presented or published research papers.

We expect that this research results will have a significant impact on the understanding of inter- effects between tactical actions and components of their environment. We also expect that the approaches will facilitate the decision making and planning processes, as related to tactical actions which are dynamically affected by unraveling events in the environment of their operation.

Task 4

The work carried out under tasks described above has resulted in additions and modifications to the tool suite developed for the research. The following are some of the features added to TEMPER reflecting either the implementations of new algorithms or fixes done to old ones to enhance computational performance.

1. A new set of inference algorithms.
2. A heuristic based branch-and-bound algorithm for the solution of General Temporal Problem.
3. The hierarchical temporal planning module.
4. A new interface for the application. This adds Gantt chart and other graphical I/O support to the application.

5. A new set of influence and temporal models/algorithms for Influence Networks and Activation Influence Networks in *Pythia*.

An application TEMPER on criminal forensics was carried out during the reporting period. The information regarding events surrounding some criminal activity or an act of terrorism unfolds in no specific order. The information gathered, in turn, may be incomplete, partially specified, and possibly inaccurate, or inconsistent, making it difficult for investigators and counter-terrorism experts to piece together the events that can help resolve some of the investigative questions. The time-sensitive information, the information about the timing of events surrounding a criminal/terrorist act, may contain hidden patterns or temporal relations that can help identify missing links in an investigation. This calls for a formal, computer-aided approach to such an analysis. The study found TEMPER to be a promising approach/tool for such a forensic analysis. In the study, a set of temporal facts was taken from the London bombing incident that took place on July 7, 2005, to illustrate the application. The information used in the illustration was gathered through the online news sites. A hypothetical investigation on the information was carried out to identify certain time intervals of potential interest to crime investigators. A paper written on the application was nominated for the "Best Student Paper" award in the 2006 Command and Control Research and Technology Symposium (CCRTS). A published version of the paper is provided in Section 9.4.

Task 5

This was a six month effort that concluded in September 2005. A detailed report on this task is given in Section 9.3 of this report. A brief description of the results is given as follows:

The development and maturation of Network Centric Operations is one of the two major dimensions of an Information Age Transformation of the DoD. The other dimension, the mission space, a space that represents the full range of the operations a force must be able to successfully undertaken, is being transformed as well. The 21st century mission space encompasses a wide range of operations including civil-military operations that require (1) an effects-based approach to operations and (2) the ability to work effectively in coalition environments that include not only other militaries but also other government entities, international organizations, and a variety of non-governmental and private voluntary organizations (NGOs and PVOs). Network Centric Operations require the coevolution of concepts of operation, approaches to command and control (including organization, doctrine, and C2 and information processes) with a robustly networked force, and their materiel and systems. Planning is an integral part of command and control processes, and thus needs to be "reinvented" in order to leverage the capabilities of a robustly networked force and be compatible with network-centric concepts of operation. Thus, moving to Network Centric Operations involves a redefinition of command arrangements and processes, including the adoption of effects-based planning, better integration of planning and execution, and a redefinition of the nature of mission participants and their respective roles, responsibilities, and interactions.

Transformation is by definition more than incremental improvements or sustaining innovations. Transformation requires venturing beyond one's comfort zones to explore new concepts of operation, new approaches to command and control, and new processes.

As such, it would be unreasonable to expect the answers to be apparent or the data for analysis to be available. The way ahead involves the formulation, design, and implementation of a campaign of experimentation and an associated program of research focused on the development and assessment of interactive and dynamic effects-based planning in the context of 21st century Network Centric Operations.

This research effort found that there was an urgent need for a campaign of research and experimentation focused on developing a network-centric approach to air and space command and control, specifically the development and assessment of approaches to mission planning in a network-centric environment. Having concluded that such a campaign of experimentation is necessary, this document provides the intellectual foundation for such a campaign. It provides appropriate definitions for key concepts, a conceptual reference model for network-centric, effects-based planning and execution, identifies a set of research issues, and identifies key activities that are on the critical path to transforming the planning and execution of air and space operations.

3. ACCOMPLISHMENTS/NEW FINDINGS

The progress made during the period includes: a) Enhancements to the PIL approach and implementation of new algorithms in TEMPER together with rigorous testing and debugging of the software; b) Redesigning of the TEMPER application for Gantt chart and other graphical I/O interfaces; c) An introductory framework for integrating qualitative and quantitative spatial and temporal information; d) An exploratory study of the structure and contents involved in a capability package and a possible use of ontologies to map the two notions of capability and requirements; e) A promising application of the tool, TEMPER, for criminal forensics and for analyzing acts of terrorism; f) Reformulation and expansion of Influence Network modeling approach; g) A new modeling approach called Activation Timed Influence Nets, and h) Providing an intellectual foundation for a campaign of experimentation focused on developing a network-centric approach to air and space command and control.

4. PERSONNEL SUPPORTED

Faculty:

Dr. Abbas K. Zaidi
Dr. David Alberts
Prof. Alexander H. Levis
Dr. Lee W. Wagenhals

Graduate Student(s) (PhD)

Mr. Abdul Qadar Kara
Mr. Sajjad Haider

Graduate Student(s) (MS)

Mr. Mashhood Ishaque
Ms. Shanthi Ramaswami
Ms. Juan Luo

5. PUBLICATIONS (Entire period)

- ‡ Published Paper in Research Journal/Special Issue
- † Published Paper in Conference Proceedings
- Thesis / Dissertation / Technical Report

‡ Mashhood Ishaque, Abbas K. Zaidi, "Project Management Using Point Graphs," *Journal of Systems Engineering*, 12 (1), To appear in 2009. **[See Section 9.2]**

† Titsa Papantoni-Kazakos, Abbas K. Zaidi, Muhammad F. Rafi, "An Algorithm for Activation Timed Influence Nets," The 2008 IEEE International Conference on Information Reuse and Integration, IEEE SMC, Las Vegas, 8, Submitted, July 2008. **[See Section 9.7]**

‡ Abbas K. Zaidi, Faisal Mansoor, P. Papantoni-Kazakos, "Theory of Influence Networks," *IEEE Transactions on SMC, Part A: Systems and Humans*, Revision Submitted, 2008. **[See Section 9.6]**

† Mashhood Ishaque, Faisal Mansoor, Abbas K. Zaidi, "An Inference Mechanism for Point-Interval Logic," The 21st International FLAIRS Conference, Association for the Advancement of Artificial Intelligence, Coconut Grove, FL, May 2008. **[See Section 9.3]**

† Abbas K. Zaidi, Faisal Mansoor, P. Papantoni-Kazakos, "Modeling with Influence Networks Using Influence Constants: A New Approach," 2007 IEEE International Conference on Systems, Man, and Cybernetics, IEEE Systems, Man, and Cybernetics, Montreal, Canada, 6, 2007. **[An extended version sent for publication (Section 9.6)]**

† Abbas K. Zaidi, Mashhood Ishaque and Alexander H. Levis, "Project Management Using Point Graphs," *Conference on Systems Engineering Research (CSER)*, INCOSE, 2007. **[An extended version accepted for publication (Sections 9.1 and 9.2)]**

Abbas K. Zaidi and Alexander H. Levis, "Using Temporal Reasoning for Criminal Forensics against Terrorists," *Descartes Conference on Mathematical Models in Counterterrorism*, Center for Advanced Defense Studies, Unpublished, 2006.

‡† Abbas K. Zaidi, Mashhood Ishaque and Alexander H. Levis, "Using Temporal Reasoning for Criminal Forensics against Terrorists," In Newton Howard and Ammar Qusaibaty (Eds.), *Mathematical Models for Counterterrorism, Springer Series*, 2007. **[See Section 9.4]**

† Abbas K. Zaidi, Mashhood Ishaque and Alexander H. Levis, "On Applying Point-Interval Logic to Criminal Forensics," 2006 Command and Control Research and Technology Symposium (CCRTS), 2006. **[An extended version was later published (Section 9.4)]**

‡ Abbas K. Zaidi and Lee Wagenhals, "Planning Temporal Events using Point Interval Logic," Special Issue, Mathematical and Computer Modeling, (43), 1229-1253, 2006. **[See Section 9.1]**

† Abbas K. Zaidi, Lee Wagenhals and Sajjad Haider, "Assessment of Effects Based Operations Using Temporal Logic," in Proc. of The 10th International Command and Control Research and Technology Symposium, McLean VA. June 2005. **[See Section 9.5]**

† Abbas K. Zaidi and Mashhood Ishaque "Time Sensitive Planning Using Point-interval Logic," in Proc. of The 10th International Command and Control Research and Technology Symposium, McLean VA. June 2005. **[An extended version accepted for publication (Section 9.2)]**

† Abbas K. Zaidi, Sajjad Haider and Alexander H. Levis, "On Temporal Analysis of Timed Influence Nets using Point Graphs," in Proceedings of The 18th International FLAIRS Conference, FL. 16 May 2005. **[A revised version appeared in another conference proceeding (Section 9.5)]**

- Mashhood Ishaque, "On Temporal Planning and Reasoning with Point-Interval Logic," MS Thesis, Department of Compute Science, George Mason University. **[Submitted with 2006 Progress Report]**
- David S. Alberts, "Planning for Network Centric Operations." September 2005. **[See Section 9.8]**

6. INTERACTIONS/TRANSITIONS

a. Participation in Conferences, Meetings:

- Participation and presentation at the International IEEE Conference on SMC, 2007, Montreal, Canada.
- Participation and presentation at the annual program review, Optimization and Discrete Mathematics, AFOSR, 2007.
- Participation and presentation of a paper at the 'Conference on Systems Engineering Research,' March 2007, NJ
- Participation and presentation of a paper at the 'Conference on Systems Engineering Research,' April 2006, CA
- Participation and presentations in the meetings of the research personnel for AFOSR multi-university research initiative titled: Model and Systems Integration Technology for the C2 Wind-tunnel (WIN), March 27, 2006 (Kickoff) and May 11, 2006 (Group Planning Meeting.)
- Participation (by invitation) in the 'AFOSR Workshop on Methods for Designing, Planning, and Operating Systems of Systems,' Indianapolis, IN, May 17-18, 2006.
- Participation and presentation at the annual program review, Optimization and Discrete Mathematics, AFOSR, May 22-24, 2006
- Participation and presentation of a paper at the '2006 Command and Control Research and Technology Symposium (CCRTS),' June 19-20, 2006.
- 18th International FLAIRS Conference, Florida, May 2005.
- 10th International Command and Control Research and Technology Symposium, McLean Virginia, June 2005.
- Program Review, Optimization and Discrete Mathematics, AFOSR, August 22-24, 2005.

b. Consultative and Advisory Functions

None at this time.

d. Transitions:

The API of TEMPER has been successfully embedded in *Pythia* for temporal assessment of COAs. The Pythia application is being tested by an Air Force Intelligence organization (NASIC) and is also used by NPS and JIEDDO.

The standalone application TEMPER is being tested/explored by Dr. Susan Numrich at the Institute for Defense Analyses (IDA), Alexandria, VA, for research projects/grants management and planning purposes.

The Pythia application is being extended by the inclusion of new influence and temporal models developed in the last year of the effort. The version with new models and algorithms is due for release in Fall 2008.

7. NEW DISCOVERIES, INVENTIONS, PATENT DISCLOSURES:

None

8. HONORS/AWARDS

Prof Alexander H. Levis: AFCEA Special Merit Award, 2006.

Prof. Alexander H. Levis: Lifetime Achievement Award at 10th International Command and Control Research and Technology Symposium, McLean Virginia, June 2005.

9. ATTACHMENTS

9.1 Planning Temporal Events using Point Interval Logic

9.2 Project Management Using Point Graphs

9.3 An Inference Mechanism for Point-Interval Logic

9.4 Using Temporal Reasoning for Criminal Forensics against Terrorists

9.5 Assessment of Effects Based Operations Using Temporal Logic

9.6 Theory of Influence Networks

9.7 An Algorithm for Activation Timed Influence Nets

9.8 Planning for Network Centric Operations

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Mathematical and Computer Modelling 43 (2006) 1229–1253
MATHEMATICAL AND COMPUTER MODELLING
www.elsevier.com/locate/mcm

Planning temporal events using point–interval logic

Abbas K. Zaidi*, Lee W. Wagenhals[1]

System Architectures Lab, Center of Excellence in C3I, George Mason University, Fairfax, VA 22030, USA

Received 10 March 2005; accepted 4 May 2005

Abstract

The paper presents a temporal logic and its application to planning time-critical missions. An extended version of the Point–Interval Logic (PIL) is presented that incorporates both point and interval descriptions of time. The points and intervals in this formalism represent time stamps and time delays, respectively, associated with events/activities in a mission as constraints on or as resultants of a planning process. The lexicon of the logic offers the flexibility of qualitative and/or quantitative descriptions of temporal relationships between points and intervals of a system. The provision for qualitative temporal relationships makes the approach suitable for situations where all the required quantitative information may not be available to planners. A graph-based approach, called the Point Graph (PG) methodology, is shown to implement the axiomatic system of PIL by transforming the temporal specifications into Point Graphs. A temporal inference engine uses the Point Graph representation to infer and verify the feasibility of temporal relations among system intervals/points. The paper demonstrates the application of PIL and its inference engine to a mission-planning problem.

Keywords: Point–Interval Logic; Point Graphs; Planning; Critical path analysis

1. Introduction

The growing need for a formal logic of time for modeling and analyzing real world systems has led to the emergence of various kinds of representations and reasoning schemes for temporal information. The earliest attempts at formalizing a time calculus date back to 1941 by Findlay [1], and 1955 by Prior [2]. Since then, there have been a number of attempts on issues related to this subject matter, like the topology of time [3–9], and first-order and modal approaches to time [10–22], treatments of time for simulating action and language. Several attempts at mechanizing the temporal reasoning processes have also been reported in the literature [22–31]. The list of references provided here is not at all exhaustive and may have missed some of the major contributions. Some of the important sources for interested readers are Stock [32], Galton [3], Shoham [33], Gabbay et al. [34], Anger et al. [35], Barringer et al. [36], the TIME workshop series [37–42], the Proceedings of the International Conference on Temporal Logic [43,44], and the Proceedings of the Workshop on Spatial and Temporal Reasoning [45]. Artale and Franconi [46], Augusto [47],

* Corresponding address: MSN 4B5, Science and Technology II, George Mason University, 4400 University Drive, Fairfax, VA 22030, USA. Tel.: +1 703 993 1774; fax: +1 703 993 1601.
E-mail addresses: szaidi@ieee.org (A.K. Zaidi), lwagenha@gmu.edu (L.W. Wagenhals).
[1] Tel.: +1 703 993 1712; fax: +1 703 993 1601.

0895-7177/$ - see front matter
doi:10.1016/j.mcm.2005.05.018

and Ma and Knight [48] have presented surveys on some special classes of temporal reasoning research in their papers. The development of some of these formalisms has matured enough to attract complexity analyses for the computational aspects of these calculi and their subclasses [49–55]. A majority of the available temporal approaches were formulated to handle issues relevant to a specific application area of interest to the scholar(s) who proposed the approach. Therefore, each approach comes with its advantages in addressing certain issues and disadvantage of lack of generality in handling others. Artificial intelligence, software engineering, planning, cognitive sciences, time-sensitive databases, program verification and modeling of discrete event systems are some of the examples of the application areas which have contributed centrally to the formulation of time calculi.

The work presented in this paper is motivated by an effort to integrate temporal and spatial aspects of a large-scale discrete event system (DES) for dynamic control (i.e., planning and replanning) purposes. The dynamic planning problems require satisfaction of two types of constraints, namely temporal and spatial, in a potentially volatile environment. The time-sensitive aspect requires a planner to sequence time intervals (points) associated with mission activities without violating any of the system specifications, given a priori and/or during execution. The spatial aspect, on the other hand, needs to look at the (spatial) availability of the physical resources capable of handling the required task list. This paper presents an approach for addressing the time aspect only. The underlying approach for handling temporal information can be nontrivially extended to incorporate spatial information. This issue is not discussed in this paper. The challenge that remains to be addressed is integrating the two types of constraints in a single analytical formulation of the problem and devising an engine that can: (a) model the two aspects using a single integrated formalism; (b) validate the feasibility of all types of constraints; and (c) help generate a feasible plan, given additional mission requirements. In addition, the constraints may not be all quantitatively and/or fully specified; therefore, there is a further need to have provisions for both quantitative and qualitative constraint handling. This paper, in this context, addresses some of these issues only for the time-sensitive aspects of the dynamic control problem. A number of other researchers have also attempted to use temporal reasoning formalisms for planning, plan merging, conditional planning, and planning with uncertainty problems [24,56–64].

The point–interval formalism (PIL), presented in this paper, originated from an earlier work on temporal knowledge representation and reasoning by Zaidi [66]. The earlier version of PIL was an extension of Hamblin's time primitives [67] and Allen's Interval logic [18–20]. (Similar extensions to Allen's ontology have also been reported by several other researchers [48,51,68].) This extension allowed the provision of points (i.e., intervals with zero length) in Allen's ontology. The inclusion of a point as a primitive entity was done for the reason that a system's temporal aspects might be represented in terms of properties that hold for certain time intervals, processes taking some time to complete, but can also be represented as events/occurrences requiring virtually no time to take place, e.g. instantaneous events. The formalism presented an axiomatic system for the point–interval logic. A Petri net [69, 70] model was shown to represent this axiomatic system by transforming the system's specifications (i.e. qualitative temporal relations between system entities) given by statements of point–interval logic into Petri net structures. The Petri net structure, with some of its analytical tools, was subsequently renamed a Point Graph (PG). The temporal version of the point–interval formalism was called point–interval temporal logic (PITL). A temporal inference engine based on this Point Graph representation infers new temporal relations among system intervals, identifies temporal ambiguities and errors (if present) in the system's specifications, and finally identifies the intervals of interest defined by the user. The inference engine was also implemented in a tool, called TEMPER, for a subclass of PITL, called Single Timeline Single Future (STSF) systems [66]. A follow-up paper by Zaidi and Levis, in 2001 [71], extended the point–interval approach one step further by adding provisions for 'date/clock' times and time 'distances' for points and intervals. This extension allowed the assignment of actual lengths to intervals, time distances between points, and time stamps to points representing the actual time of occurrences, whenever such information is available. The paper, therefore, addressed the issue of combining qualitative and quantitative information into a single formalism, as posed by Guesgen [72]. The paper also presented the modified tool, TEMPER-II, which implements the inference engine for both qualitative and quantitative temporal information.

The dynamic control of DESs often requires revising a temporal model produced during and/or after a system specification phase, e.g., the constraints or system/mission requirements may change during or before a plan's execution. The earlier implementation of TEMPER did not support revision (add, modify, delete); even a minor change could not be made unless one re-edited the temporal statements and restarted TEMPER's engine from scratch. This limitation made it infeasible for use in planning real time systems and other dynamic application areas, especially with large temporal inputs. This weakness in the Point Graph formalism was overcome in [73]. The changes in the

temporal specifications are classified into 'local', 'regional', and 'global' based on the impact of the change on the original set [61]. The approach in [73] uses a multi-layered PG structure to keep the input specifications in the lowest layer of a PG. The qualitative and quantitative information available to TEMPER is processed and kept at a higher layer in the PG. The inference engine works at the higher layer to answer queries and infer new temporal relations; however, a change in the inputs is processed at the lowest layer and its 'effects' are propagated upwards. The affected parts of the PG determine the kind of change that TEMPER is required to accommodate.

The contribution of this paper can be summarized as follows. The paper consolidates the results of the previous work on a point–interval logic and its application to model and plan time-sensitive aspects of a mission. The point–interval formalism presented in this paper is an extension to an earlier approach [66,71]. The extension allows for a larger class of temporal systems to be handled by incorporating an enhanced input lexicon that captures both qualitative and quantitative temporal information, representation of flexibility in temporal specifications, an improved verification and inference mechanism, and a suite of analysis tools. The logic presented, although independently developed, can be considered an extension to the Gerevini and Schubert's *timegraphs* [74] with the added provision for metric information. The language of the logic is shown to be expressive enough for handling time-sensitive aspects of events/activities in a plan. It allows for the specification of temporal relations between points, points and intervals, and *partially ordered* temporal relations between intervals. The classical scheduling approaches, e.g., the Critical Path Method (CPM) and Project Evaluation and Review Technique (PERT) [65], only allow modeling of duration-based activities (intervals) and specification of *strictly ordered* temporal relations between the intervals. For example, given two intervals X and Y, the only temporal relations that can be specified between the two are X precedes Y, Y precedes X, or there is no relation between the two. For cases where a temporal constraint needs to be specified between the end points of the two activities, e.g., the start of X precedes the start (or end) of Y, etc., the conventional approaches provide us with no mechanisms for handling the temporal situations. The logic presented in this paper not only offers a more expressive input language for specification of temporal relations between points and/or intervals, but its graph-theoretic knowledge representation and the inference mechanism also overcome the limitations of the classical approaches. The approach presented, therefore, offers an enhanced formalism for planning in terms of its expressive language for specifications, provision for point and interval descriptions of temporal events, and a powerful inference engine.

The contents of the paper are organized as follows. The extended Point–Interval Logic (PIL) and its axiomatic system are presented in Section 2. Section 3 introduces the Point Graphs for knowledge representation and a reasoning mechanism for the logic. The issue of verification of PIL statements is discussed in Section 4. The point–interval formalism is shown to handle temporal information in Section 5 with the help of a temporal lexicon for the PIL statements. This section also presents a planning application of the formalism that identifies the *critical* activities, time slacks for the non-critical activities, and offers a graph-based tool for 'what-if' analysis of the plan. The illustration is supported with a small, but non-trivial, real world example. The paper concludes in Section 6 with a brief discussion on future directions and problems to resolve.

2. Point–Interval Logic (PIL)

2.1. Lexicon

The lexicon of the Point–Interval Logic (PIL) consists of the following primitive symbols:

Points (Event): A point X is represented as [pX, pX] or simply [pX].

Intervals: An interval X is represented as [sX, eX], where 'sX' and 'eX' are the two end points of the interval, denoting the 'start' and 'end' of the interval, s.t. sX < eX. (In the sequel, the term interval is used to refer to both intervals and points, if not explicitly stated otherwise.)

Point relations: These are the relations that can exist between two points. The set of relations R_P is given as: $R_P = \{<, =\}$.

Interval relations: These are the *atomic* relations that can exist between two intervals. The set of relations R_I is given as[2]: $R_I = \{<, m, o, s, d, f, =\}$.

[2] For a temporal system, relation '<' corresponds to '*Before*', 'm' to '*Meets*', and so forth. See Table 6 for a complete list.

Point–interval relations: These are the *atomic* relations that can exist between a point and an interval. The set of relations R_π is given as: $R_\pi = \{<, s, d, f\}$.

Functions: The following two functions are used to represent quantitative information associated with intervals.

The interval length function assigns a non-zero positive real number to a system interval, e.g.,

$$\text{Length X} = d, \quad \text{where } X = [sX, eX], d \in \mathfrak{R}^+.$$

The stamp function assigns a real number to a system point, e.g., Stamp p1 = t, $t \in \mathfrak{R}$.

Proposition 2.1. *The PIL relations in sets* R_P, R_I, *and* R_π *are mutually exclusive and exhaustive, i.e.,*

(a) *if* 'X Ri Y'*,* Ri *is a PIL relation, then there does not exist another PIL relation* Rj *such that* 'X Rj Y' *also holds true;*

(b) *for any two intervals (points)* X *and* Y *there must exist an PIL relation* Ri *such that either* 'X Ri Y' *or* 'Y Ri X' *holds true (with the exception of the '=' relation where* 'X = Y' *is equivalent to* 'Y = X'*).*

Note: The second part of the proposition only holds for a *complete* system of PIL statements [66].

2.2. Syntactic and semantic structure

The syntactic and semantic structure of atomic relations in PIL is shown in Table 1. The table outlines three possible cases (i.e., interval–interval, point–interval, and point–point) and the corresponding semantically relevant relations that can exist between points and/or intervals, represented by generic symbols X and Y. A qualitative relation between two intervals can be described with the help of algebraic inequalities, also shown in Table 1, among points representing the start and end of these intervals. The readers are cautioned on the dual use of some of these symbols for representing both algebraic and PIL relations. The context of their use makes the distinction very clear and the different uses of the same symbol in two different contexts should not be confused with each other.

A system of PIL statements is given by a conjunction of statements each describing a PIL relation between a *unique* pair of intervals/points. Example 2.1 presents two syntactically correct systems of PIL statements.

Example 2.1.

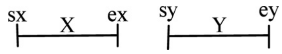
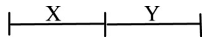
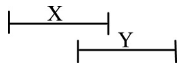
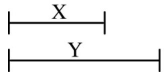
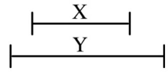
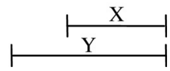
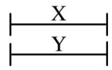
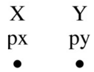
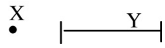
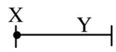
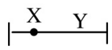
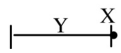
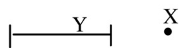
Δ1:	event 1 s Process 1	Δ2:	X o Y
	Stamp event 1 = 1000		Length [sX, sY] = 10
	Length Process 1 = 10		Length [sY, eX] = 8
	Event 2 f Process 1		Z o Y
	Event 2 s Process 2		Length [sZ, sY] = 5
	Length Process 2 = 20		Length [sY, eZ] = 8

Two points, p1 and p2, on a real number line are related to each other by one of the following three algebraic relations: '<' (less/greater than), '=' (equal to), and '≤' (less/greater than or equal to). A relation Ri between two intervals X and Y, denoted as 'X Ri Y' can, therefore, be represented as a four-symbol string made of elements from the alphabet $\{<, =, >, \leq, \geq, ?\}$, where the first (leftmost) symbol represents the algebraic relation between sX and sY, the second symbol that between sX and eY, the third symbol the relation between eX and sY, and the fourth that between eX and eY. The '?' is added to incorporate incomplete information. Table 2 shows this string representation for each of the atomic PIL relations.

The provision of the '≤' (and '≥') relation between two points in the string representation of Table 2 results in the definition of *compound* relations between points and intervals.

Definition 2.1 (*Compound PIL Relation*). A compound PIL relation between a pair of intervals (or between an interval and a point) is defined to be a disjunction of two or more atomic PIL relations between the two intervals, i.e., $(X < m\ Y) = (X < Y) \vee (X\ m\ Y)$.

Table 1
Expressions in PIL and their semantics

Qualitative Relations			
CASE I — X and Y both intervals with non-zero lengths:			
	X = [sX, eX], Y = [sY, eY]		
1.	X < Y	eX < sY	
2.	X m Y	eX = sY	
3.	X o Y	sX < sY; sY < eX; eX < eY	
4.	X s Y	sX = sY; eX < eY	
5.	X d Y	sX > sY; eX < eY	
6.	X f Y	sY < sX; eY = eX	
7.	X = Y	sX = sY; eX = eY	
CASE II—X and Y both points: X = [pX] and Y = [pY]			
1.	X < Y	pX < pY	
2.	X = Y	pX = pY	
CASE III— X is a point and Y is an interval: X = [pX] and Y = [sY, eY]			
1.	X < Y	pX < sY	
2.	X s Y	pX = sY	
3.	X d Y	sY < pX < eY	
4.	X f Y	pX = eY	
5.	Y < X	eY < pX	
Quantitative Relations			
X, Y are points, Z is an interval and d is a real value			
1. Stamp X = d		2. Length [X, Y] = d	3. Length Z = d

The approach presented in this paper, however, does not allow all possible disjunctive combinations of PIL relations between intervals and points. The only allowable disjunctive combinations of PIL relations that can be used to construct compound relations are given in Tables 3–5. The definition of the allowable relations between intervals is done by the use of symbols from the alphabet $\{<, =, >, \leq, \geq, ?\}$ in constructing the string representation of a relation. For the sake of brevity, Table 3 does not show the remaining inverse PIL relations (Definition 2.2) and their corresponding string representations.

Definition 2.2 (*Inverse Relation*).

(a) Let Ri be an atomic PIL relation. The inverse of Ri, denoted by Ri^{-1}, between two intervals X and Y, represented as X Ri^{-1} Y, is defined to be equivalent to Y Ri X.

Table 2
Analytical representation of PIL relation

CASE I— X and Y both intervals with non-zero lengths:				
X = [sX, eX], Y = [sY, eY]				
X Ri Y	sX Vs. sY	sX Vs. eY	eX Vs. sY	eX Vs. eY
X < Y	<	<	<	<
X m Y	<	<	=	<
X o Y	<	<	>	>
X s Y	=	<	>	<
X d Y	>	<	>	<
X f Y	>	<	>	=
X = Y	=	<	>	=
X unknown Y	?	?	?	?
CASE II—X and Y both points:				
X = [pX] and Y = [pY]				
X Ri Y	pX Vs. pY			
X < Y	<			
X = Y	=			
X ? Y	?			
CASE III—X is a point and Y is an interval:				
X = [pX] and Y = [sY, eY]				
X Ri Y	pX Vs. sY	pX Vs. eY		
X < Y	<	<		
X s Y	=	<		
X d Y	>	<		
X f Y	>	=		
Y < X	>	>		
X ? Y	?	?		

(b) The inverse of an inverse results in the atomic PIL relation, i.e., $(\mathrm{Ri}^{-1})^{-1} = \mathrm{Ri}$.

(c) Let ρ be a compound PIL relation. The inverse of ρ, denoted as ρ^{-1}, is obtained by inverting all the constituent atomic relations in ρ, i.e., $(\mathrm{mof}^{-1})^{-1}$ is equal to $\mathrm{m}^{-1}\mathrm{o}^{-1}\mathrm{f}$.

2.3. Axiomatic system

The inference mechanism of PIL uses the analytical representation of PIL statements presented in Tables 2–5 and the following axioms to infer unknown relations among system intervals. The axioms have been generated by an exhaustive enumeration of all possibilities involving points and/or intervals.

A. Point axioms [66]

Let p1, p2, and p3 be points defined on a real number line.

(A.1) $(\mathrm{p1} < \mathrm{p2}) \wedge (\mathrm{p3} < \mathrm{p1}) \rightarrow (\mathrm{p3} < \mathrm{p2})$
(A.2) $(\mathrm{p1} < \mathrm{p2}) \wedge (\mathrm{p3} = \mathrm{p1}) \rightarrow (\mathrm{p3} < \mathrm{p2})$
(A.3) $(\mathrm{p1} < \mathrm{p2}) \wedge (\mathrm{p3} \leq \mathrm{p1}) \rightarrow (\mathrm{p3} < \mathrm{p2})$
(A.4) $(\mathrm{p1} < \mathrm{p2}) \wedge (\mathrm{p2} = \mathrm{p3}) \rightarrow (\mathrm{p1} < \mathrm{p3})$
(A.5) $(\mathrm{p1} < \mathrm{p2}) \wedge (\mathrm{p2} \leq \mathrm{p3}) \rightarrow (\mathrm{p1} < \mathrm{p3})$
(A.6) $(\mathrm{p1} = \mathrm{p2}) \wedge (\mathrm{p3} = \mathrm{p1}) \rightarrow (\mathrm{p3} = \mathrm{p2})$
(A.7) $(\mathrm{p1} = \mathrm{p2}) \wedge (\mathrm{p3} \leq \mathrm{p1}) \rightarrow (\mathrm{p3} \leq \mathrm{p2})$
(A.8) $(\mathrm{p1} = \mathrm{p2}) \wedge (\mathrm{p2} \leq \mathrm{p3}) \rightarrow (\mathrm{p1} \leq \mathrm{p3})$
(A.9) $(\mathrm{p1} \leq \mathrm{p2}) \wedge (\mathrm{p3} \leq \mathrm{p1}) \rightarrow (\mathrm{p3} \leq \mathrm{p2})$
(A.10) $(\mathrm{p1\_p2}) \wedge (\mathrm{p3\_p1}) \rightarrow (\mathrm{p3}\ ?\ \mathrm{p2})$

(The symbol '?' represents unknown relation. The symbol '_' is used to denote remaining combinations of the relation, $\{<, =, >, \leq, \geq, ?\}$, not covered by axioms 1–10).

Table 3
Analytical representation of allowable PIL relations between two intervals

sX Vs. sY	sX Vs. eY	eX Vs. sY	eX Vs. eY	X Ri Y
$<$	$<$	$<$	$<$	$<$
$<$	$<$	$=$	$<$	m
$<$	$<$	$>$	$<$	o
$<$	$<$	$>$	$\leq$	of^{-1}
$<$	$<$	$>$	?	$\mathrm{od}^{-1}\mathrm{f}^{-1}$
$<$	$<$	$\leq$	$<$	$<\mathrm{m}$
$<$	$<$	$\geq$	$<$	mo
$<$	$<$	$\geq$	$\leq$	mof^{-1}
$<$	$<$	$\geq$	?	$\mathrm{mod}^{-1}\mathrm{f}^{-1}$
$<$	$<$	?	$<$	$<\mathrm{mo}$
$<$	$<$	?	$\leq$	$<\mathrm{mof}^{-1}$
$<$	$<$	?	?	$<\mathrm{mod}^{-1}\mathrm{f}^{-1}$
$=$	$<$	$>$	$<$	s
$=$	$<$	$>$	$=$	$=$
$=$	$<$	$>$	$\leq$	$\mathrm{s}=$
$=$	$<$	$>$	?	$\mathrm{ss}^{-1} =$
$>$	$<$	$>$	$<$	d
$>$	$<$	$>$	$=$	f
$>$	$<$	$>$	$\leq$	df
$\leq$	$<$	$>$	$<$	os
$\leq$	$<$	$>$	$\leq$	$\mathrm{osf}^{-1} =$
$\leq$	$<$	$>$	?	$\mathrm{oss}^{-1}\mathrm{d}^{-1}\mathrm{f}^{-1} =$
$\leq$	$<$	$\geq$	$<$	mos
$\leq$	$<$	$\geq$	$\leq$	$\mathrm{mosf}^{-1} =$
$\leq$	$<$	$\geq$	?	$\mathrm{moss}^{-1}\mathrm{d}^{-1}\mathrm{f}^{-1} =$
$\leq$	$<$	?	$<$	$<\mathrm{mos}$
$\leq$	$<$	?	$\leq$	$<\mathrm{mosf}^{-1} =$
$\leq$	$<$	?	?	$<\mathrm{moss}^{-1}\mathrm{d}^{-1}\mathrm{f}^{-1} =$
$\geq$	$<$	$>$	$<$	sd
$\geq$	$<$	$>$	$=$	$\mathrm{f}=$
$\geq$	$<$	$>$	$\leq$	$\mathrm{sdf}=$
?	$<$	$>$	$<$	osd
?	$<$	$>$	$=$	$\mathrm{ff}^{-1} =$
?	$<$	$>$	$\leq$	$\mathrm{osdff}^{-1} =$
?	$<$	$>$	?	$\mathrm{oo}^{-1}\mathrm{ss}^{-1}\mathrm{dd}^{-1}\mathrm{ff}^{-1} =$
?	$<$	$\geq$	$<$	mosd
?	$<$	$\geq$	$\leq$	$\mathrm{mosdff}^{-1} =$
?	$<$	$\geq$	?	$\mathrm{moo}^{-1}\mathrm{ss}^{-1}\mathrm{dd}^{-1}\mathrm{ff}^{-1} =$
?	$<$	?	$<$	$<\mathrm{mosd}$
?	$<$	?	$\leq$	$<\mathrm{mosdff}^{-1} =$
?	$<$	?	?	$<\mathrm{moo}^{-1}\mathrm{ss}^{-1}\mathrm{dd}^{-1}\mathrm{ff}^{-1} =$
?	$\leq$	$\geq$	?	$\mathrm{mm}^{-1}\mathrm{oo}^{-1}\mathrm{ss}^{-1}\mathrm{dd}^{-1}\mathrm{ff}^{-1} =$
?	$\leq$	?	?	$<\mathrm{mm}^{-1}\mathrm{oo}^{-1}\mathrm{ss}^{-1}\mathrm{dd}^{-1}\mathrm{ff}^{-1} =$
?	?	?	?	$\ll^{-1}\mathrm{mm}^{-1}\mathrm{oo}^{-1}\mathrm{ss}^{-1}\mathrm{dd}^{-1}\mathrm{ff}^{-1} =$

B. Interval axioms

Let X and Y be intervals; X = [sX, eX] and Y = [sY, eY].

(B.1) $(\mathrm{sX} < \mathrm{sY}) \to (\mathrm{sX} < \mathrm{eY})$
(B.2) $(\mathrm{sX} = \mathrm{sY}) \to (\mathrm{sX} < \mathrm{eY}) \wedge (\mathrm{eX} > \mathrm{sY})$
(B.3) $(\mathrm{sX} \leq \mathrm{sY}) \to (\mathrm{sX} < \mathrm{eY})$
(B.4) $(\mathrm{sX} > \mathrm{sY}) \to (\mathrm{eX} > \mathrm{sY})$
(B.5) $(\mathrm{sX} \geq \mathrm{sY}) \to (\mathrm{eX} > \mathrm{sY})$
(B.6) $(\mathrm{sX} > \mathrm{eY}) \to (\mathrm{sX} > \mathrm{sY}) \wedge (\mathrm{eX} > \mathrm{sY}) \wedge (\mathrm{eX} > \mathrm{eY})$
(B.7) $(\mathrm{sX} = \mathrm{eY}) \to (\mathrm{sX} > \mathrm{sY}) \wedge (\mathrm{eX} > \mathrm{sY}) \wedge (\mathrm{eX} > \mathrm{eY})$

Table 4
Analytical representation of allowable PIL relations between two points

pX Vs. pY	X Ri Y
$<$	$<$
$=$	$=$
$>$	$<^{-1}$
$\leq$	$<=$
$\geq$	$=<^{-1}$
?	$<=<^{-1}$

Table 5
Analytical representation of allowable PIL relations between a point and an interval

pX Vs. sY	pX Vs. eY	X Ri Y
$<$	$<$	$<$
$=$	$<$	s
$>$	$<$	d
$>$	$=$	f
$>$	$>$	$<^{-1}$
$>$	$\leq$	df
$>$	$\geq$	f$<^{-1}$
$>$	?	df$<^{-1}$
$\leq$	$<$	$<$s
$\geq$	$<$	sd
$\geq$	$\leq$	sdf
$\geq$	?	sdf$<^{-1}$
?	$<$	$<$sd
?	$\leq$	$<$sdf
?	?	$<$sdf$<^{-1}$

(B.8) $(sX \geq eY) \rightarrow (sX > sY) \wedge (eX > sY) \wedge (eX > eY)$
(B.9) $(eX < sY) \rightarrow (sX < sY) \wedge (sX < eY) \wedge (eX < eY)$
(B.10) $(eX = sY) \rightarrow (sX < sY) \wedge (sX < eY) \wedge (eX < eY)$
(B.11) $(eX \leq sY) \rightarrow (sX < sY) \wedge (sX < eY) \wedge (eX < eY)$
(B.12) $(eX < eY) \rightarrow (sX < eY)$
(B.13) $(eX = eY) \rightarrow (sX < eY) \wedge (eX > sY)$
(B.14) $(eX \leq eY) \rightarrow (sX < eY)$
(B.15) $(eX > eY) \rightarrow (eX > sY)$
(B.16) $(eX \geq eY) \rightarrow (eX > sY)$

C. Point–interval axioms

Let X be a point and Y an interval; X = [pX] and Y = [sY, eY].

(C.1) $(pX < sY) \rightarrow (pX < eY)$
(C.2) $(pX = sY) \rightarrow (pX < eY)$
(C.3) $(pX \leq sY) \rightarrow (pX < eY)$
(C.4) $(pX > eY) \rightarrow (pX > sY)$
(C.5) $(pX = eY) \rightarrow (pX > sY)$
(C.6) $(pX \geq eY) \rightarrow (pX > sY)$

The inference mechanism of PIL constructs the analytical representation for the pairs of intervals with unknown relations with the help of the axioms. The resulting string representation of the relation(s) is pattern matched with the string representations of Tables 2–5 to infer possible relation(s) between the intervals. An inference engine for PIL, therefore, requires an exhaustive enumeration of the result through all feasible combinations of available statements, provided no knowledge of the system's correctness is available a priori [66]. An inference engine that outputs the

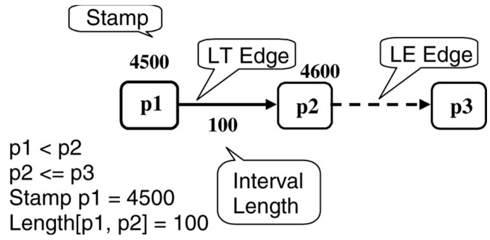


Fig. 1. Point Graph representation of a set of PIL expressions.

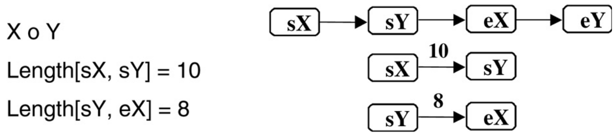


Fig. 2. PIL statement to PG translation.

result as soon as it finds the first feasible set of inputs can only be applied to a known consistent system of PIL statements. This, in turn, requires a front-end verification mechanism for the PIL statements. Another point to note is that the axiomatic system presented in this section does not take into account the quantitative information that might be available to the system. Zaidi, in 1999 [66], proposed a graph-based methodology, termed the Point Graphs approach, to resolve these problems. A discussion on this methodology follows in the next section.

3. Point Graphs (PG)

The inference mechanism of PIL is implemented with the help of a graph, called the Point Graph (PG). The expressions in PIL are transformed to their PG representations, and the graph so constructed is processed before being used for the inferences. This section presents a detailed account of the PG representation and the graph operations applied to it.

Definition 3.1 (*Point Graph*). A Point Graph, PG (V, E_A, D, T), is a directed graph with:

V: Set of vertices with each node or vertex $v \in V$ representing a point on the real number line. Points pi, pj, ..., pn are represented as a composite point [pi; pj; ...; pn] if all are mapped to a single point on the line.

E_A: Union of two sets of edges: $E_A = E \cup E_{\leq}$, where

- E: Set of edges with each edge $e_{12} \in E$, between two vertices v1 and v2, also denoted as (v1, v2), representing a relation '<' between the two vertices—($v1 < v2$). The edges in this set are called LT edges;
- $E_{\leq}$: Set of edges with each edge $e_{12} \in E_{\leq}$, between two vertices v1 and v2, also denoted as (v1, v2), representing a relation '≤' between the two vertices—($v1 \leq v2$). The edges in this set are called LE edges.

D: Edge-length function (possibly partial): $E \to \mathfrak{R}^+$

T: Vertex-stamp function (possibly partial): $V \to \mathfrak{R}$.

Fig. 1 presents a three-node Point Graph with vertex stamps and arc length, and the corresponding PIL system represented by the PG. The figure also presents a correspondence between the stamps and edge lengths: a PG with only stamps can be represented by an equivalent PG with edge length expressions and vice versa by using a reference stamp for the conversion.

A relation Ri between two intervals X and Y can now be represented by an equivalent Point Graph representation by translating the algebraic inequalities shown in Tables 3–5 to corresponding PGs. Fig. 2 illustrates the conversion with the help of some example PIL statements and their corresponding PGs.

The PG representing the entire system of PIL statement is then constructed by unifying (Definition 3.3, below) individual PGs to a (possibly) single connected graph. The unifying process only looks at the labels of the nodes to identify equalities, and does not take into consideration the arc lengths assigned to edges in the PG.

(a) Branch. (b) Join.

Fig. 3. Branch and join nodes in Point Graphs.

Definition 3.2 (*Pre-set (Post-set)*). A pre-set (post-set) of a node contains all the nodes in V that have directed edges originating from (terminating at) them and terminating at (originating from) node v. The notation *v (v*) represents the pre-set (post-set) of a node v.

$$\forall\ \mathrm{vi},\ \mathrm{vi} \in {}^{*}\mathrm{v}, \quad \text{then } (\mathrm{vi}, \mathrm{v}) \in \mathrm{E_A}.$$

Similarly,

$$\forall\ \mathrm{vi},\ \mathrm{vi} \in \mathrm{v}^{*}, \quad \text{then } (\mathrm{v}, \mathrm{vi}) \in \mathrm{E_A}.$$

Definition 3.3 (*Unification*).

(a) Let vi = [pi; . . . ; pn] and vj = [pj; . . . ; pm] be two nodes in a PG representation. If there exists a point pk such that pk ∈ [pi; . . . ; pn] and pk ∈ [pj; . . . ; pm] or T(vi) = T(vj) then the two nodes are merged into a single composite node 'vi;vj' such that

$$\mathrm{vi};\ \mathrm{vj} = [\mathrm{pi}; \ldots; \mathrm{pn}] \cup [\mathrm{pj}; \ldots; \mathrm{pm}]$$
$${}^{*}(\mathrm{vi};\ \mathrm{vj}) = {}^{*}\mathrm{vi} \cup {}^{*}\mathrm{vj}$$
$$(\mathrm{vi};\ \mathrm{vj})^{*} = \mathrm{vi}^{*} \cup \mathrm{vj}^{*}.$$

The change in pre-sets and post-sets of unified nodes results in redefinition of the set E_A in the PG representation. The nature of the edges involved in the unification does not change in the redefinition:

$$\mathrm{T}(\mathrm{vi};\ \mathrm{vj}) = \mathrm{T}(\mathrm{vi}) = \mathrm{T}(\mathrm{vj})$$

or:

(b) For all vi and vj ∈ V, s.t. T(vi) < T(vj,) construct a directed edge from node vi to vj with D(vi, vj) = T(vj) − T(vi). (The corresponding sets V, E_A, and the functions D, T, are accordingly updated.)

The unified PG is then scanned for join and branch nodes (Definition 3.4, below) with quantitative information on their incoming and outgoing edges, respectively. The PG is then folded (Definitions 3.5 and 3.6, below) at these types of nodes. The folding process establishes new relations among system intervals, inferred through the quantitative analysis of the known relations specified by interval lengths and stamps.

Definition 3.4 (*Branch (Join) Node*). A vertex v ∈ V in a Point Graph is termed a branch (join) node if it has multiple outgoing (incoming) edges connected to it.

Fig. 3 shows a pictorial representation of a branch and a join node in Point Graphs.

Definition 3.5 (*Branch Folding*). A branch node vi ∈ V is said to be folded if, for all vj and vk in the post-set of vi, we have:

(a) D(vi, vj) < D(vi, vk) the edge from vi to vk, denoted as (vi, vk), is replaced by an edge (vj, vk) with

$$\mathrm{D}(\mathrm{vj},\ \mathrm{vk}) = \mathrm{D}(\mathrm{vi},\ \mathrm{vk}) - D(\mathrm{vi},\ \mathrm{vj})$$

and the vertex vk removed from the post-set
or

(b) D(vi, vj) = D(vi, vk), the two vertices vj and vk are merged into a single vertex with composite label ‘vj;vk’, and

$$D(vi, vj; vk) = D(vi, vk)\{= D(vi, vj)\}$$

or

(c) vi has multiple edges to vj; if the edges are all of the same type (LT or LE) then only one edge is retained and others are deleted; if at least one of them is of type LT then it is retained and others are deleted; if D(vi, vj) is defined for one of these edges, the value is assigned to the surviving edge.
(The corresponding sets V, E_A, and the functions T, D are accordingly updated.)

The methodology applies the branch folding process to all the original and newly created (formed during the folding process) branch nodes in the unified net. The branch folding process, when applied to all the branch nodes of a graph, yields a partially folded PG having nodes with at most one outgoing edge with edge-length expression. Since all the edges in the PG may not have edge lengths associated with them, the branch folding may not result in a branch-node-free PG. A join folding process, which applies a similar process to all the joins in the graph, further treats the PG so obtained.

Definition 3.6 (*Join Folding*). A join node vi ∈ V is said to be folded if, for all vj and vk in the pre-set of vi, we have:

(a) D(vj, vi) < D(vk, vi), the edge (vk, vi), is replaced by an edge (vk, vj) with

$$D(vk, vj) = D(vk, vi) - D(vj, vi)$$

and the vertex vk removed from the pre-set
or

(b) D(vj, vi) = D(vk, vi), the two vertices vj and vk are merged into a single vertex with composite label ‘vj;vk’, and

$$D(vj; vk, vi) = D(vk, vi)\{= D(vj, vi)\}$$

or

(c) vi has multiple edges from vj; if the edges are all of the same type (LT or LE) then only one edge is retained and others are deleted; if at least one of them is of type LT then it is retained and others are deleted; if D(vj, vi) is defined for one of these edges, the value is assigned to the surviving edge.
(The corresponding sets V, E_A, and the functions T, D are accordingly updated.)

A single application of join folding after a single application of branch folding is all that is needed to fully fold the graph. A proposition by Zaidi and Levis, in 2001 [71], ensures the fact that single applications of branch folding followed by join folding are enough to fold the graph completely (the term ‘completely’ is used relative to the quantitative information available in the PG).

Fig. 4 illustrates the process of converting a set of PIL statements to their PG representation (Fig. 4(a), and 4(b)). The figure also shows the result of the unification of the PG (Fig. 4(c)). The join folding and branch folding of the PG are shown in parts (d) and (e), Fig. 4.

The PG representation of PIL statements helps the inference mechanism of PIL to construct the string representation for the pairs of intervals (Tables 3–5) with unknown relations by performing a simple search in the PG constructed after unification and folding processes. The existence of a directed path from a node ‘p’ to another ‘q’ with at least one LT edge in it establishes the relation ‘p < q’ between the two points. A path between the two nodes with only LE type edges establishes the relation ‘p ≤ q’ between the two. An inference for a PIL relation between two intervals requires at most eight searches to be performed, two for each pair of start/end points. The resulting string representation is pattern matched with the strings in Tables 3–5 to identify the corresponding atomic/compound PIL relation. As mentioned earlier, an inference resulting in a compound relation of the type $RiRjRk^{-1}$ between two intervals X and Y represents the disjunction of ‘X Ri Y’, ‘X Rj Y’, and ‘Y Rk X’. The search for the directed path between two vertices in a PG uses a depth-first search with arc lengths as the heuristic measure; the depth-first search engine first explores the outgoing edge of the current vertex with a length expression. The search, therefore, finds the path between two vertices that has (possibly) all its constituent edges with length expressions. The sum of all these lengths gives the total distance between the two vertices (points). Similarly, if the stamp of one of these points is known, the stamp of the other can be calculated by adding or subtracting the distance (path length) between the two.

The illustration in Fig. 4 shows the new PIL relations that can be inferred for the PIL system modeled by the approach. The PIL statements ‘Z f X’ and ‘Length [sX, sZ] = 5’ can easily be inferred through the PG in Fig. 4(e).

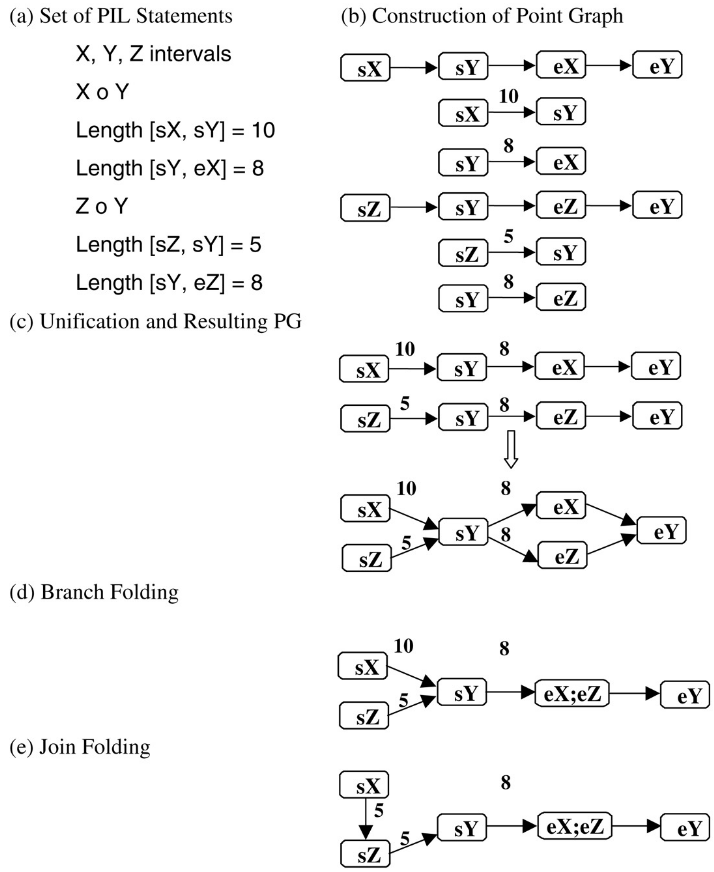


Fig. 4. Steps in PG construction.

4. Verification of PIL statements

The inference mechanism described in Section 3 may result in erroneous and inconsistent results provided the system of PIL statements, represented by the PG, contains *inconsistent* information. The inference, on the other hand, is guaranteed to yield valid assertions given a consistent PIL system and corresponding PG representation. This section characterizes the inconsistencies in a PIL system and in its PG representation. The section also presents methods for verifying a PIL system for these erroneous instances.

Definition 4.1 (*Inconsistency [75]*). A set of statements (inferences) is said to be inconsistent if the statements in the set cannot all be true at the same time.

Definition 4.2 (*Interpretation Function,* I_f). An interpretation function in PIL assigns a stamp to each node in the PG representation of a system of PIL statements.

I_f: $V_\Delta \rightarrow \mathfrak{R}$, where Δ is a system of PIL statements.

Definition 4.3 (*Interpretation, I*). An interpretation of a system of PIL statements Δ assigns a 'True' or 'False' value to each atomic PIL statement in Δ.

I: $\Delta \rightarrow$ {True, False}.

Theorem 4.1. *For any interpretation function for a set of PIL statements* Δ*, there is a unique interpretation of* Δ.

The theorem follows from the definitions of interpretation function, interpretation, and the analytical representation of PIL relations given in Tables 3–5.

Definition 4.4 (*Satisfaction, Model*). A system of PIL statements Δ is satisfied by an interpretation I iff $I(\Delta) = \text{True}$, also denoted as $|=_I \Delta$. An interpretation satisfying a system of PIL statements Δ is called a Model of Δ.

Theorem 4.2. *A system of PIL statements Δ is inconsistent if and only if it is unsatisfiable, i.e., there does not exist an interpretation that satisfies Δ.*

The theorem directly follows from the definition of inconsistency (Definition 4.1) and definition of an interpretation of a system of PIL statements. The definition of inconsistency in Definition 4.1, Theorem 4.2, and Proposition 2.1 lead to another (operational) characterization of inconsistency in Theorem 4.3. The theorem is an extension of an earlier result presented by Zaidi and Levis [71].

Theorem 4.3 (*Inconsistency in PIL*). *A system's description in PIL contains inconsistent information iff*

(a) *for some intervals* X *and* Y*, and atomic PIL relations* Ri *and* Rj*, both '*X Ri Y*' and '*X Rj Y*',* $i \neq j$, *or '*X Ri Y*' and '*Y Rj X*' (with the exception of = relation) hold true;*
or
(b) *for some intervals and/or points, the system can determine two string representations such that at least one pair of the algebraic inequalities representing relationships between the corresponding points represents an inconsistency; let the two string representations be '*abcd*' and '*uvwx*', where* a, b, c, d, u, v, w, *and* $x \in \{<, =, >, \leq, \geq, ?\}$*; one of the (unordered) pairs of corresponding inequalities, i.e.,*

$$(a, u), (b, v), (c, w), \text{ or } (d, x) \in \{(<, =), (<, >), (<, \geq), (=, >), (>, \leq)\};$$

or
(c) *for a point* p1*, the system calculates two different stamps;*
or
(d) *for some points* p1 *and* p2*, '*$p1 < p2$*', the system can determine two different lengths for the interval* [p1, p2].

The part (a) of the theorem entails part (b), but not vice versa. It is therefore imperative to look for the cases described by part (b) for identification of inconsistent PIL statements. Some of the inconsistent cases, of the type defined in the other two parts (c) and (d), are trivially detected during the unification process: whenever two nodes with different stamps are merged into a single node—an inconsistency.

Once a unified Point Graph representation is achieved, the graph is checked for other inconsistent cases defined by the part (b) in the theorem. Such inconsistent cases are characterized by the following theorem.

Theorem 4.4 (*[66]*). *A set of PIL statements is inconsistent if the PG representation of the set contains self-loops and/or cycles with some LT type edges involved in the cycles.*

A necessary condition for a consistent set of temporal statements is, therefore, given as:

Theorem 4.5 (*[66]*). *A set of temporal statements is consistent only if the PG representation of the set is an acyclical graph.*

Proofs. A system of PIL statements is inconsistent if two (or more) *different* PIL relations can be established and/or inferred between two intervals X and Y. The corresponding string representation (Tables 3–5) for the two relations would mean that two different inequalities could be established between at least one pair of ending points (e.g., p1 and p2) of the two intervals. In PG representation, the two different inequalities would result in two paths, one from p1 to p2 and the other from p2 to p1. The two paths together form a cycle. □

The verification mechanism of PG representation identifies these inconsistent cases by applying the following result (Theorem 4.6).

Theorem 4.6 (*[66]*). *A Point Graph contains cycles if and only if it has non-zero S-invariants calculated for the Connectivity matrix of the Point Graph.*

Definition 4.5 (*Connectivity Matrix*). A Point Graph with n directed edges and m nodes can be represented by a $(n \times m)$ matrix J, the Connectivity matrix. The rows correspond to edges and the columns correspond to nodes.

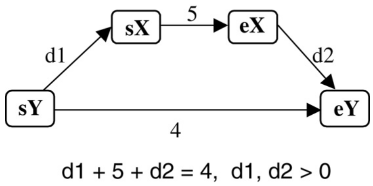


Fig. 5. Example inconsistent cases.

- $\underline{j}_{ij} = 1$ if the directed edge in ith row originates from the jth node,
- $\underline{j}_{ij} = -1$ if the directed edge in ith row terminates in the jth node,
- $\underline{j}_{ij} = 0$ if the directed edge in ith row is not connected to jth node.

Note that in constructing the Connectivity matrix no distinction is made between LT and LE type edges.

Definition 4.6 (*S-Invariant*). Given the Connectivity matrix J of a Point Graph, an S-invariant is an $n \times 1$ non-negative integer vector X of the kernel of J^T, i.e.,

$$J^T X = 0.$$

The rows of the matrix X correspond to the edges in the PG. The set of edges corresponding to the non-zero elements in an S-invariant, represented as <X>, has been shown in [66] to reveal a directed elementary circuit in the PG.

The verification approach, therefore, constructs a Connectivity matrix of the unified PG and calculates the S-invariants of the graph. The S-invariants can be calculated using an improved version of Farkas algorithm by [76, 77]. The resulting non-zero S-invariants identify the cycles (inconsistencies) in the system. Once cycles are detected in a PG by calculating non-zero S-invariants, the nodes responsible for these cycles can be easily identified. This will, in turn, identify intervals involved in these cycles. This information can be used to correct the system of PIL statements.

The folding process (Definitions 3.5 and 3.6) establishes new PIL relations, among system intervals, inferred through the quantitative analysis of the known relations specified by interval lengths and stamps. The possible inconsistencies present in the quantitative inputs may hinder the folding process or result in erroneous structures [71] of the folded graph. The type of inconsistency defined by Theorem 4.3(d) may reveal itself during the folding process: if during folding a PG the process finds multiple edges between a branch (join) node and a vertex in its post-set(pre-set), where these edges have different lengths associated with them, then the process halts and reports an error. The inconsistency can also result in creation of new cycles in the graph during the folding process. These cycles can be identified using another application of S-invariant algorithm.

The creation of cycles during the folding process can have serious effects on the graph. Once a cycle is created during the folding process, it tends to attract the remaining vertices in the PG towards itself. And if the PG has edge lengths on all its edges, the folding process ends up with a folded PG, which has a single cycle with all its vertices collapsed into it. The phenomenon is termed the ‘Black Hole Effect’ [71]. The intensive computational effort required in folding a PG, and a subsequent loss of it due to the black hole effect demand an earlier detection of cycles during the folding process itself. The folding procedure is, therefore, tailored to identify cycles by assigning dummy time stamps to vertices being folded: reassignment of a time stamp to an already marked vertex prompts the presence of a cycle. A folded PG with leftover branch and join nodes should also be checked for multiple directed paths from any branch node to any other join node. The length expressions corresponding to each such path are equated to each other and the resulting set of equations is checked for feasibility. A set of infeasible equations signals an inconsistent case present in the system. Fig. 5 presents an example of such an inconsistent case.

A technique alternative to the S-invariant algorithm, called the Path-searching algorithm, given by Ma [61], uses the Adjacency (Definition 4.7, below) and Reachability (Definition 4.8, below) matrices of the PG representation to uncover the cycles and inconsistent multiple paths between pairs of nodes (Theorem 4.3(d)). The Path-searching algorithm employs techniques by Busacker and Saaty [78] and Warshall’s algorithm [79] to identify the erroneous cases. A recent implementation of the verification mechanism is done by employing the Path-searching algorithm on the unified PG representation of the input system. The following is a brief description of the Path-searching algorithm.

Definition 4.7 (*Adjacency Matrix*). A Point Graph with n directed edges and m nodes can be represented by an $(m \times m)$ matrix A, the Connectivity matrix. An element a_{ij} in the matrix represents the number of edges from node 'i' to node 'j' in the PG.

Definition 4.8 (*Reachability Matrix*). Given a simple digraph $G = (V, E)$ with vertices indexed as $V(G) = \{v1, \ldots, vn\}$, the matrix R with rank n, $R = (r_{ij})_{n \times n}$, is called the Reachability matrix, where

$$r_{ij} = \begin{cases} 1 & \text{there is at least one path between vi and vj} \\ 0 & \text{there is no path between vi and vj.} \end{cases}$$

In particular,

$$r_{ii} = \begin{cases} 1 & \text{there is a cycle from vi and vi} \\ 0 & \text{otherwise.} \end{cases}$$

The Reachability matrix is a 0–1 (Boolean) matrix, and matrix addition and multiplication can be adapted to Boolean addition and Boolean multiplication. Warshall's algorithm [79] employs the idea and provides an efficient mechanism for calculating the Reachability matrix from the Adjacency matrix of a Directed Graph. The algorithm has a time complexity of $O(n^3)$, where n is number of nodes in the graph. Using the Reachability matrix, checking the connectivity between any two nodes vi and vj becomes very easy: $r_{ij} = 1$ means there is at least one path from vi to vj; otherwise, $r_{ij} = 0$. If a node vi is involved in a cycle, the element r_{ii} in R must take the value of 1. Thus, if there are non-zero elements on the diagonal of matrix R for a Point Graph, this Point Graph has cycles in it. The Path-searching algorithm is used to detect and identify both cycles and inconsistent paths between pairs of branch and join nodes in a unified Point Graph. A description of the algorithm is provided in Definition 4.9.

Definition 4.9 (*Path-Searching Algorithm [61]*). Given the Adjacency matrix A, the Reachability matrix R, and a pair of a source node vi and a target node vj, the output of the algorithm is a list of paths L from the source node to the target node vj. The following steps produce the list of paths L from the inputs:

1. Check the connectivity from node vi to vj using matrix R. If $r_{ij} > 0$, then there is at least one path from vi to vj; otherwise exit.
2. Construct X(0), a row vector, by picking the row in A that corresponds to the node vi.
3. Iterate for $k = 0$ to $n - 1$, where n is the number of nodes, and X(k) is not a zero vector:
   a. Calculate the elements of $X(k + 1)$ via the dot product of X(k) with each column of matrix A.
4. Iterate for $m = 0$ to k:
   a. If $X(m)j > 0$, where X(m)j is the number of paths from vi to vj of length $m + 1$.
      i. Find all nodes to which vi has a path of length m and from which vj is reachable through a single edge. This is obtained by taking AND of $X(m - 1)$ and the transpose of jth column of A. This determines the last edge for each of these paths of length $m + 1$. For each of these nodes r, find those nodes to which vi has a path of length $m - 1$ and from which r is reachable through a single edge. This identifies the second edge and this goes on until all the edges for each of these paths are identified.
      ii. Append all these paths to the list L.

The inconsistent paths are identified by running the algorithm for each pair of a branch and a join node in a PG. The length of each path in the list L corresponds to an algebraic expression. The expressions for all the paths in L are equated together to identify inconsistent lengths.

The Path-searching algorithm is followed by the folding (Definitions 3.5 and 3.6) of the PG. The folded PG is used for making inferences. The technical details on the implementation and the accompanying software application are available on the website of System Architectures Lab, George Mason University, at http://viking.gmu.edu.

5. Application to temporal systems

This section presents an application of PIL for modeling temporal situations. Table 6 lists the PIL relations, function names, and their corresponding temporal lexicon. The table also suggests some high-level temporal relations that can be used to represent compound PIL relations. The task of designing a comprehensive and suitable language

Table 6
Temporal equivalents of PIL relations

PIL relation/function name	Corresponding temporal equivalent
Atomic relations	
$<$	Before
m	Meets
o	Overlaps
s	Starts
d	During
f	Finishes
$=$	Equals
Stamp	Time
Length	Length
Some compound relations	
$<=$ (for points only)	
$<$s (for point and interval)	
$<$m (for intervals only)	Precedes
osd	Ends_During
$\mathrm{o^{-1}df}$	Starts_During
$\mathrm{ss^{-1}} =$	Starts_With
$\mathrm{ff^{-1}} =$	Ends_With
$<\mathrm{mod^{-1}f^{-1}}$	Starts_Before
$<$mo	Starts_Before_Starts
$<$mosd	Ends_Before_Ends
$<\mathrm{moo^{-1}sdd^{-1}ff^{-1}} =$	Starts_Before_Ends

for temporal relations is left as a choice for the user, who may define his/her own (natural language) constructs for the entire set of compound relations given in Tables 3–5. A system of temporal statements can, therefore, be constructed using the temporal lexicon with the PIL syntax. The temporal version of the logic is termed Point–Interval Temporal Logic (PITL). Once the temporal inputs are specified using the new lexicon, the rest of the formalism is identical to the approach presented in the previous sections. It can obviously be concluded that the notions of interpretation, satisfiability, inconsistency, and inference in PIL are equivalent to the corresponding temporal interpretation, temporal satisfiability, temporal inconsistency, and temporal inference in PITL. In this section, we take the temporal implementation of PIL a step further by introducing a suite of PG-based temporal analyses for a possible application to mission-planning problems.

5.1. Mission planning

This section presents a subclass of PITL (Definition 5.1) for modeling temporal requirements and/or constraints of a mission to be planned. The points and intervals of the logic correspond to time stamps and time delays, respectively, associated with events/activities in the mission as constraints on or as resultants of a planning process. The lexicon of the logic offers the flexibility of both qualitative and quantitative descriptions of temporal relationships between points and intervals of the system. The definition of the subclass (Definition 5.1), however, puts some restrictions on the type of temporal information that can (or cannot) be handled by the analysis presented in this section. (Note that the restrictions in Definition 5.1 do not apply to the approach presented in the previous sections; a generic system of PITL statements can be processed by the methods presented and the inference mechanism of PITL can be invoked to infer unknown temporal relationships between system intervals.)

Definition 5.1 (*Subclass of PITL,* $\mathtt{A}$). The subclass $\mathtt{A}$ of PIL is described with the help of the following requirement on a system of PIL statements Δ:

A system of PITL statements $\Delta \in \mathtt{A}$ if in the string representation of statements in Δ, every strict inequality ($<$ or $>$) between two points p1 and p2 is accompanied by a length expression for the distance between the two points. The following is a set of necessary conditions for the PIL statements that follow this characterization:

1. all intervals defined in the system are provided with their lengths, i.e., ∀ X, where X = [sX, eX], 'Length X = d' ∈ Δ, for some d ∈ $\mathfrak{R}$;
2. for a pair of points, X and Y, if 'X Before Y' ∈ Δ, then 'Length[X, Y] = d' ∈ Δ, for some d ∈ $\mathfrak{R}$;
3. for a point X and an interval Y, if 'X Before Y' ∈ Δ, then 'Length[X, sY] = d' ∈ Δ, for some d ∈ $\mathfrak{R}$;
4. for a point X and an interval Y, if 'Y Before X' ∈ Δ, then 'Length[eY, X] = d' ∈ Δ, for some d ∈ $\mathfrak{R}$;
5. for a pair of intervals, X and Y, if 'X Before Y', 'X Overlaps Y', or 'X During Y' ∈ Δ, then:

   'Length[sX, sY] = d', 'Length[sX, eY] = d', 'Length[sY, eX] = d', or
   'Length[eX, eY] = d' ∈ Δ, for some d ∈ $\mathfrak{R}$;

Corollary 5.1 (*PG Representation of the Subclass*). *The unified and folded PG representing a system in* $\mathbb{A}$ *has a total edge-length function.*

The corollary signifies the type of constraints that can be used to model the temporal relations between mission activities. It requires that a strict '<' (Before) relation between any two points (representing point activities and/or start/end of interval activities) be accompanied by a length function. This is more of a requirement for the application of algorithms that will be presented in this section than on the type of temporal systems that can be modeled. A temporal system that does not conform to Definition 5.1 can be pre-processed, without violating any temporal requirements, using the following steps: Replace every relation of the type '$X < Y$' between two points X and Y, by relations '$X < D$', 'Length[X, Z] = d', and '$Z \leq Y$', where Z is a dummy activity and d is a user-defined smallest time increment, e.g. for systems with only integer lengths and time stamps d = 1.

Once a mission's requirements are converted to PITL statements, the temporal system is then converted to its PG representation (Definition 3.1). The PG, so obtained, is processed by applying unification and folding processes (Definitions 3.3, 3.5 and 3.6). The unified PG is checked for inconsistency by the approach presented in Section 4. The verification of PG either reports infeasible temporal requirements in the input, or ensures the fact that the input PITL system is satisfiable. The inference mechanism of the logic can now be invoked to determine temporal relations between intervals/points of interest. In order to construct a model (Definition 4.4) of the temporal system, the PG is added with a pair of *source* and *sink* nodes (Definition 5.2). At this point, an *optimized* model of the PITL system can be constructed by solving the mathematical program defined in Definition 5.3 for the PG representation. The model is termed optimized for the reason that it constructs an interpretation of the system with the minimized start-to-end ($V_{out} - V_{in}$) time duration. Alternatively, a graph-based analysis can be used to construct a similar model of the temporal system with the added benefit of a plan representation that can be analyzed for alternatives. The time stamps on individual nodes are not considered in the two approaches; the stamps can be ignored without any loss of generality. The time stamp can be easily incorporated either before or after the analysis that follows. Once a plan is constructed using the approach, it be shifted on a timeline to match with the stamps provided in the input PITL statements.

Definition 5.2 (*Source and Sink Nodes to PG*). A source node V_{in} and a sink V_{out} node are added to the PG representation of a system of PITL statements by applying the following:

(a) ∀ vi, vi ∈ V such that $^*v = \phi$ (i.e., null set), connect the source node V_{in} to all vi's by LE type edges (V_{in}, vi);
(b) ∀ vi, vi ∈ V such that $v^* = \phi$, connect the sink node V_{out} to all vi's by LE type edges (vi, V_{out}).

Definition 5.3 (*Mathematical Program Representing PG*). Given a PG ($V \cup \{V_{in}, V_{out}\}$, E_A, D, T), where $E_A = E \cup E_{\leq}$, a mathematical program for constructing an interpretation of the PITL represented by PG is defined as:

Objective Function:

$$\text{Minimize } V_{out} - V_{in}$$

Subject to:

$$vj - vi = D(vi, vj), \quad \forall\, (vi, vj) \in E$$
$$vi \leq vj, \quad \forall\, (vi, vj) \in E_{\leq}$$
$$vi \geq 0, \quad \forall\, vi \in V.$$

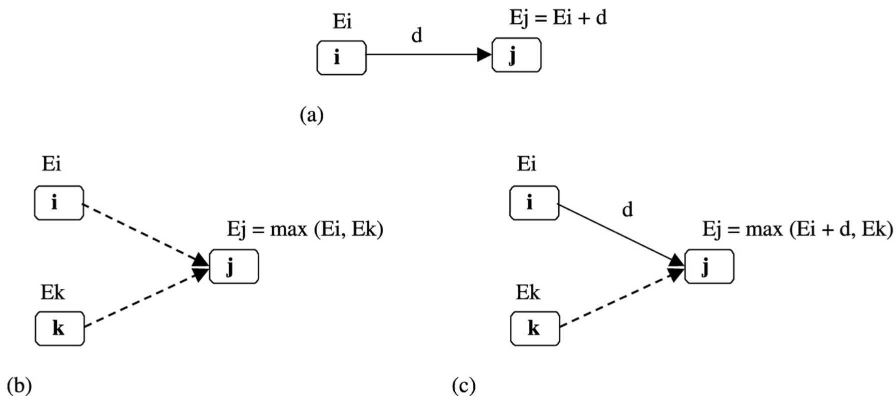


Fig. 6. Illustration of forward pass.

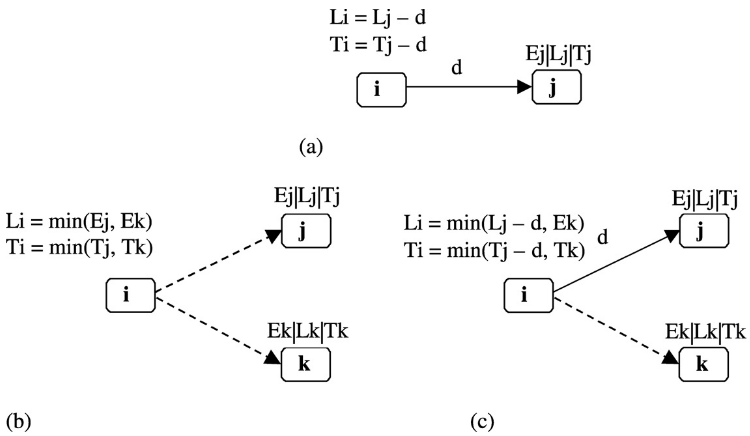


Fig. 7. Illustration of reverse pass.

The graph-based approach assigns three parameters to each node in the PG representation. The parameter values are calculated by running an analysis on the graph. The values of these parameters help determine the *critical activities* (Definition 5.9, below) and time floats/slacks (Definition 5.10, below) for intervals in the system, and *interval/point activities* (Definitions 5.7 and 5.8) defined for the PG under consideration. The three parameters are termed as *earliest occurrence (Ev), late occurrence (Lv), and latest occurrence (Tv)* of a node 'v', and are formally defined in Definitions 5.4–5.6. The analysis applies two passes through the PG representation. The first, forward pass (Definition 5.4), calculates the value for the earliest occurrence time of a node; the other, reverse pass (Definitions 5.5 and 5.6), calculates the values for the late and latest occurrences of a node in the PG. Figs. 6 and 7 illustrate the two passes with the help of example cases.

Definition 5.4 (*Earliest Occurrence of a Node, Ev, in PG — Forward Pass*). The earliest occurrence Ev of a node v, v $\in$ V, is defined to be the smallest time stamp on the node that satisfies the earliest occurrences of the preceding nodes, i.e.,

Let $^*\mathrm{v} = \{\mathrm{vi}\}$

$$\mathrm{Ev} = \begin{cases} \mathrm{Evi} + \mathrm{D(vi, v)}, & \text{for (vi, v)} \in \mathrm{E} \text{ and } |{}^*\mathrm{v}| = 1 \\ \max\limits_{\mathrm{i}}[\mathrm{Evi}], & \forall\ (\mathrm{vi, v}) \in \mathrm{E}_{\leq} \\ \max\limits_{\mathrm{i}}[\mathrm{Evi}, \mathrm{Evk} + \mathrm{D(vk, v)}], & \text{for (vk, v)} \in \mathrm{E} \\ 0, & \text{otherwise.} \end{cases}$$

For a *non-critical* interval/activity [v1, v2] (Definitions 5.7–5.9), Ev1 represents the *earliest start time* of the activity.

Definition 5.5 (*Late Occurrence of a Node, Lv, in PG — Reverse Pass I*). The late occurrence Lv of a node v, v ∈ V, is defined to be the largest time stamp on the node that satisfies the earliest occurrences of the following nodes, i.e.,

Let $v^* = \{vi\}$

$$Lv = \begin{cases} Lvi - D(v, vi), & \text{for } (v, vi) \in E \text{ and } |v^*| = 1 \\ \min\limits_i [Evi], & \forall\ (v, vi) \in E_{\leq} \\ \min\limits_i [Evi, Lvk - D(v, vk)], & \text{for } (v, vk) \in E \\ Ev, & \text{otherwise.} \end{cases}$$

Definition 5.6 (*Latest Occurrence of a Node, Tv, in PG — Reverse Pass II*). The latest occurrence Tv of a node v, v ∈ V, is defined to be the largest time stamp on the node that satisfies the latest occurrences of the following nodes, i.e.,

Let $v^* = \{vi\}$

$$Tv = \begin{cases} Tvi - D(v, vi), & \text{for } (v, vi) \in E \text{ and } |v^*| = 1 \\ \min\limits_i [Tvi], & \forall\ (v, vi) \in E_{\leq} \\ \min\limits_i [Tvi, Tvk - D(v, vk)], & \text{for } (v, vk) \in E \\ Ev, & \text{otherwise.} \end{cases}$$

For a *non-critical* interval/activity [v1, v2] (Definitions 5.7–5.9), Tv2 represents the *latest completion time* of the activity.

Definition 5.7 (*Point Activity*). A node v ∈ V is called a point activity. A point, start and end points of an interval, in the PITL system, are all point activities in the PG representation of the PITL system.

Definition 5.8 (*Interval Activity*). An interval [v1, v2], where v1, v2 ∈ V, is called an interval activity if the two time points represented by the nodes v1 and v2 are the two end points of a path comprising LT type edges only.

Note that the definition of interval activities, in Definition 5.8, extends the notion of intervals in a PITL system by including composite and parts of PITL intervals to be defined as interval activities in a PG representation.

Definition 5.9 (*Critical Activity*). An activity is defined to be critical if:

(a) a delay in its start will cause a delay in the completion time of the entire mission, i.e.,
  (i) for a point activity v ∈ V, Ev = Tv;
  (ii) for an interval activity [v1, v2], where v1, v2 ∈ V, v ∈ [v1, v2], Ev = Tv
  or
(b) for an interval activity, it ‘Meets’ or is met by (Meets^{-1}) another critical activity; for a point activity, it ‘Starts’ and/or ‘Ends’ another critical activity
  or
(c) an earliest (or latest) occurrence of its start node does not ensure an earliest (or latest) occurrence of its end node, i.e.,

$$\text{for } [v1, v2],\quad Ev1 + D([v1, v2]) < Ev2, \quad \text{or}$$
$$Tv1 + D([v1, v2]) < Tv2.$$

Definition 5.10 (*Total Float (TF) and Free Float (FF)*). Total Float (TF) is the difference between the maximum time available to perform an activity and its duration. Free Float (FF) is defined by assuming that all the activities start as early as possible. It is the excess time available over its duration [80].

(a) Total float (TF) and free float (FF) for a non-critical point activity, v, are calculated from

$$TFv = Tv - Ev$$
$$FFv = Lv - Ev.$$

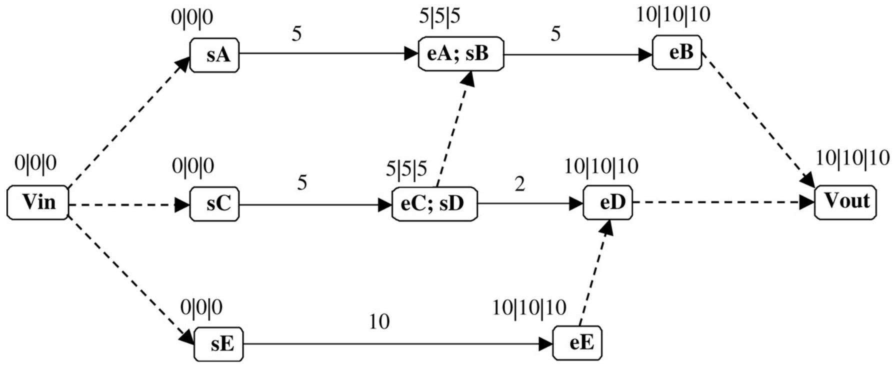


Fig. 8. PG for Example 5.1.

(b) Total float (TF) and free float (FF) for a non-critical interval activity, [v1, v2], are calculated from

$$\begin{aligned} TF_{[v1,v2]} &= Tv2 - Ev2 \\ &= Tv1 - Ev1 \\ FF_{[v1,v2]} &= Lv2 - Ev2 \\ &= Lv1 - Ev1. \end{aligned}$$

(For all critical activities TF = FF = 0.)

The condition (c) in Definition 5.9 presents an interesting and new notion of critical activities in the context of planning and scheduling literature. The condition represents an activity that, for a given start-to-end mission duration, is required to start and end at specific times, in order to satisfy the preceding and following activity timings. But, the difference between the two times, start and end, is greater than the actual duration of this activity. This difference between the actual duration and the required duration is called *stretch float (SF)*. Example 5.1 illustrates the concept with the help of an example.

Definition 5.11 (*Stretch Float (SF)*). For a critical activity [v1, v2] of type defined in Definition 5.9(c), Stretch Float (SF) is defined to be the excess time available over the duration between the earliest occurrences of its start 'v1' and end 'v2' nodes, i.e.,

$$SF_{[v1,v2]} = Ev2 - Ev1 - D([v1, v2]) \text{ [or } Tv2 - Tv1 - D(v1, v2)].$$

The stretch float, if it exists, presents the following set of alternatives to a mission planner.

(a) For a critical activity [v1, v2] with SF, any one of the following may hold:
  i. $Lv1 + D([v1, v2]) = Ev2$;
  ii. $Tv1 + D([v1, v2]) = Lv2$;
  iii. $Tv1 + D([v1, v2]) = Ev2$.
  Then, the activity is scheduled in the corresponding interval.
(b) For the activity $Tv1 + D([v1, v2]) < Ev2$ — the activity if started at the latest time still ends earlier than required by some of the preceding activities, but the activity's end time can be delayed (stretched) by an amount equal to its SF after its start. Then, the activity is stretched. (See Example 5.1, Fig. 8.)
(c) For an activity that does not satisfy any conditions in part (a) and cannot be stretched–part (b) –the mission cannot be planned without extending the start-to-end duration of the mission. The extended duration is calculated by solving the mathematical program in Definition 5.3. A dummy activity is created with length equal to the new duration (value of the objective function) and added to the list of mission activities. The analysis is applied to the new PG so obtained.

The following is a fictitious but real world example to illustrate how some of the features of PITL could be applied to military planning and execution problems. The illustration is for a precision engagement against a Time Critical Target (TCT). To do this, a scenario is presented in which several assets must concurrently perform activities with

Table 7
Mission requirements

Interval ID	Activity description	Corresponding PITL statement
A	Weapon platform ingresses to PGW launch point	Length A = 5
B	Weapon platform egresses from PGW launch point	Length B = 5
C	Target parameters are uploaded into the PGW navigation processor	Length C = 5
D	PGW is launched and flies to the target	Length D = 2
E	Local, on site activity provides navigation and guidance update to PGW	Length E = 10

Table 8
Additional constraints

Natural language description	Corresponding PITL statement
The platform will not loiter in the area due to threat considerations	A meets B
The PGW is launched immediately after the target parameters are uploaded	C meets D
The PGM launch precedes the egress	C Precedes B
Local, on site activity must cease just prior to the weapon striking the target	eE Precedes eD

implicit synchronization in order to attack a target of importance. The target is time critical in that it is difficult to locate and when it is located, it must be struck in a very short time, otherwise it will disappear. The example illustrates a small, but a non-trivial, set of temporal constraints, some of which cannot be modeled using the traditional CPM approach. The description of the example notifies the readers of such constraints.

Example 5.1. Assume the following facts and constraints apply to the planning for precision engagement of a TCT. There is a list of high value TCTs that when located and identified need to be attacked quickly with precision engagement weapons. When such a target is found, a weapon platform such as an attack aircraft must ingress to a weapon launch point to release a precision-guided weapon (PGW). During the ingress, the on-board navigation and guidance processor of the PGW will be uploaded with the precise data it needs to fly to and hit the target. During the ingress and PGW update activities, a local, on site, aid to the navigation and guidance activity must participate in providing updates to the PGW. This local, on site activity must cease just prior to the weapon striking the target. Once the weapon is launched, the launch platform egresses the area.

The mission requirements for this scenario are shown in Table 7. The table shows the five activities together with the PITL statements representing the mission operational concept. The additional constraints are described in Table 8 with their corresponding PITL statements. The constraint ‘eE Precedes eD’ (in Table 8) presents a requirement that cannot be modeled by the traditional approaches. Similarly, the constraints ‘A meets B’ and ‘C Precedes B’ together pose another such temporal requirement.

The approach presented in this paper takes the statements in Tables 7 and 8 and converts them to their corresponding PG representation. The PG is unified, verified, and folded for satisfiability. A pair of source and sink nodes is added to the PG, and forward and reverse passes are applied to the resulting PG. Fig. 8 shows the PG with all the parameter values calculated for each node in the graph. The start-to-end ($V_{out} - V_{in}$) delay of 10 time units (perhaps minutes) is the shortest possible duration for the mission to be accomplished, provided none of the constraints is violated. An inspection of the PG reveals the fact that all the activities involved are critical. The analysis also reveals that there is a stretch float (SF) condition associated with activity interval D, the PGW fly out activity. The SF suggests that the activity needs to be stretched from a duration of 2 time units to 5 time units, should the mission need to be accomplished with the minimal 10 time units with all requirements met. Further review of the scenario indicates that this activity cannot be stretched because the fly out time is fixed. Thus, the second PG shown in Fig. 9 is created by first solving the mathematical program for the constraints in the PG. Fig. 9 presents the situation where a dummy activity ‘F’ is added (see Definition 5.11(c)) to the system with ‘Length F = 13’ before recalculating the parameter values. The dummy activity is added to force the start-to-end time to be at least equal to its length of 13 time units. All the activities in Fig. 9 are critical, with their feasible time stamps underlined. The feasible time stamps are selected by first looking at the critical activities and the feasible stamps on the nodes involved. In the example case, the critical activity D (PGW fly out activity) can only start at ‘$T_{sD} = 8$’ which, in turn, makes the start of another critical activity

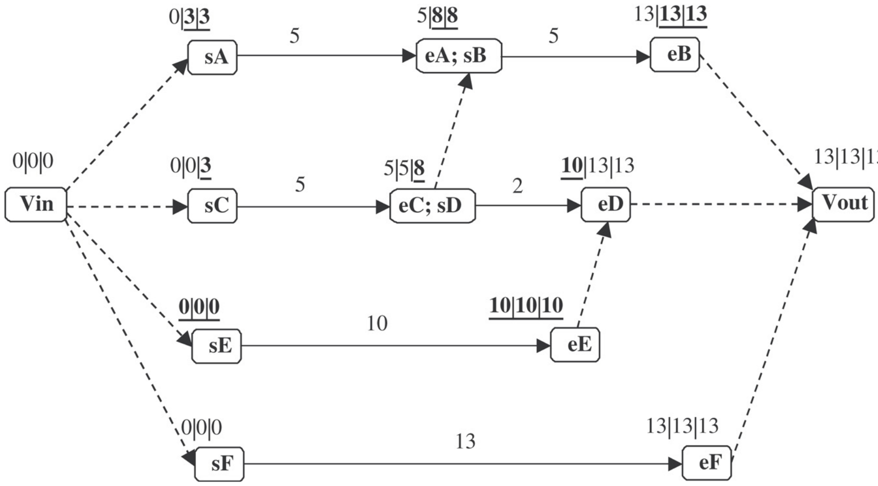


Fig. 9. Modified PG for Example 5.1.

C (uploading of target parameters) be '$T_{sC} = 3$'. Similarly, the feasible time stamps of other critical activities, i.e., A, B, and E, are selected. The values in Fig. 9, therefore, show the *only* feasible schedule for the activities involved for the mission duration of 13 time units. Thus the local, on site activity starts at time 0, the Ingress and the PGW upload start at time 3. The PGM launch occurs at time 8 and commences the Egress activity. The PGW strikes the target at time 13 just after the local, on site activity ceases. This plan provides a total mission view that can be used to provide to the individual resources that are carrying out the plan the critical start and complete times for their activities to ensure the implicit synchronization of the concurrent activities is accomplished.

This illustration is but a simple vignette. It is included to demonstrate one of many potential real world applications of the approach presented in this paper. The authors believe that the approach is capable of providing very powerful analytical capabilities to support both real world deliberate and near real time planning and plan repair problems.

Finally, the PG corresponding to a mission's requirements, with the values of the parameters calculated, can be used to construct a time chart, e.g., a Gantt chart, showing the start and finish times for each activity as well as its relationship to other activities. It also must pinpoint the critical activities. For non-critical activities the plan also must show the amount of slack or float times that can be used advantageously when such activities are delayed or when limited resources are to be used. The PG representation and the time chart can, therefore, be used for a real time and periodic control of the plan. The PG may be updated and analyzed, and, if necessary, a new plan/schedule determined for the remaining portion of the mission in a dynamic environment. An extension to the formalism that proposes an improved graph-based approach for calculating the parameter values by employing a recursive combination of the two (forward and reverse) passes will be presented in a forthcoming paper.

6. Conclusion

The paper presented a formulation of a point–interval logic and an implementation of its inference mechanism. Point Graphs are shown to implement the axiomatic system and inference mechanism of the logic. The graph-based approach is also shown to help verify the system of logic statements. The verification substantially reduces the computational effort required by the inference mechanism. The logic can be used to model time and space aspects of a system, both in terms of qualitative and quantitative information; however, the application is still under investigation for a possible integration of the two uses into a single formalism. Such an integration is expected to result in a three dimensional system with two space dimensions and a third time dimension.

The language of PIL is shown to be expressive enough for handling time-sensitive aspects of events/activities in a plan. The traditional plan/project management techniques, e.g., CPM and PERT, lack several of the constructs used in PITL for modeling temporal requirements of a plan/project. The language of PITL subsumes the class of temporal models that can be constructed by the classical approaches, and provides an enhanced lexicon for modeling instantaneous (zero-duration) events and partially ordered temporal relations between activities/events. A suite of

analyses, applied to the graph representation of PITL statements, is presented for identifying critical activities, earliest and latest times of occurrences of activities/events in a plan. The approach presented, therefore, offers an enhanced formalism for planning in terms of its expressive language for temporal specifications, provision for point and interval descriptions of temporal events, and a powerful inference engine. In that respect, it extends the capabilities of mission planners, program managers, and project designers in the operations research community for planning missions/projects beyond the traditional CPM techniques currently being used.

The formalism presented still possesses some limitations in not allowing unrestricted use of disjunctions between temporal statements. A mission (or part of it) that can be accomplished by several alternative sequences of activities cannot be effectively modeled and analyzed by the present approach. The present approach, if employed for such cases, will treat each alternative separately, resulting in a combinatorially large number of PITL systems to be modeled and analyzed by the approach.

Acknowledgements

The work was carried out with support provided by the Air Force Office of Scientific Research under contract numbers F49620-95-1-0134, F49620-98-1-0179, F49620-01-1-0008 and the Office of Naval Research under contract number N00014-93-1-0912. The authors also wish to express their gratitude to the anonymous reviewers for their valuable comments and suggestions.

References

[1] J.N. Findlay, Time: A treatment of some puzzles, Australasian Journal of Psychology and Philosophy 19 (1941) 216–235.
[2] A.N. Prior, Diodoran modalities, Philosophical Quarterly 5 (1955) 205–213.
[3] A. Galton, Temporal Logics and Their Applications, Academic Press, New York, 1987.
[4] W.H. Newton-Smith, The Structure of Time, Routledge and Kegan, London, UK, 1980.
[5] A. Bochman, Concerted instant–interval temporal semantics I: Temporal ontologies, Notre Dame Journal of Formal Logic 31 (3) (1990) 403–414.
[6] A. Bochman, Concerted instant–interval temporal semantics II: Temporal valuations and logics of change, Notre Dame Journal of Formal Logic 31 (4) (1990) 581–601.
[7] A. Galton, A critical examination of Allen's theory of action and time, Artificial Intelligence 42 (1990) 159–188.
[8] J.F.A.K. Van Benthem, The Logic of Time, second edn, Reidel, 1991.
[9] Ll. Vila, Ip: An instant–period based theory of time, in: R. Rodriguez (Ed.), Proceedings of the Workshop on Spatial and Temporal Reasoning in ECAI 94, 1994.
[10] W.V. Quine, Elementary Logic, revised ed., Harper and Row, New York, 1965.
[11] A.N. Prior, Past, Present and Future, Clarendon, Oxford, UK, 1967.
[12] N. Rescher, A. Urquhart, Temporal Logic, Springer-Verlag, Berlin, Germany, 1971.
[13] A. Pnueli, The temporal logic of programs, in: Proc. 18th IEEE Symp. Foundations Computer Science, 1977, pp. 46–67.
[14] R.P. McArthur, Tense Logic, Reidel, Dordrecht, The Netherlands, 1976.
[15] B.F. Chellas, Modal Logic. An Introduction, Cambridge University Press, 1980.
[16] Z. Manna, A. Pnueli, Verification of concurrent programs: The temporal framework, in: R.S. Boyer, J.S. Moore (Eds.), The Correctness Problem in Computer Science, Academic, London, UK, 1981, pp. 215–273.
[17] H. Reichgelt, A comparison of first order and modal logics of time, in: P. Jackson, H. Reichgelt, F. van Harmelen (Eds.), Logic Based Knowledge Representation, MIT Press, Cambridge, MA, 1989, pp. 143–176.
[18] J.F. Allen, An interval based representation of temporal knowledge, in: Proc. 7th IJCAI, 1981.
[19] J.F. Allen, Maintaining knowledge about temporal intervals, Communications of ACM 26 (1983) 832–843.
[20] J.F. Allen, Towards a general theory of action and time, Artificial Intelligence 23 (2) (1984) 123–154.
[21] J. Allen, P. Hayes, A common-sense theory of time, in: Proceedings IJCAI-85, 1, 1985, pp. 528–531.
[22] J. Allen, G. Ferguson, Actions and events in interval temporal logic, Journal of Logic and Computation 4 (1994) 531–579.
[23] K. Kahn, G. Gorry, Mechanizing temporal logic, Artificial Intelligence 9 (1977) 87–108.
[24] J. Allen, J. Koomen, Planning using a temporal world model, in: Proceedings IJCAI '83, 2, 1983.
[25] M. Abadi, Z. Manna, Nonclausal temporal deduction, in: R. Parikh (Ed.), Logics of Programs, in: Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, 1985.
[26] L. Fariñas del Cerro, Resolution modal logic, in: K.R. Apt (Ed.), Logics and Models of Concurrent Programs, Springer-Verlag, Berlin, Germany, 1985, pp. 27–56.
[27] E.M. Clarke et al., Automatic verification of finite-state concurrent systems using temporal logic specifications, ACM Transactions on Programming Languages and Systems 8 (1986) 244–263.
[28] R. Dechter, I. Meiri, J. Pearl, Temporal constraint network, Artificial Intelligence 49 (1991) 61–95.
[29] I. Meiri, Combining qualitative and quantitative constraints in temporal reasoning, in: Proc. of AAAI-91, Anaheim, CA, 1991, pp. 260–267.

[30] S. Keretho, R. Loganantharaj, Qualitative and Quantitative Time Interval Constraint Networks, ACM, 1991, pp. 239–246. 089791-382-5/91/0003/0239.
[31] Y. Yao, A Petri net model for temporal knowledge representation and reasoning, IEEE Transactions on Systems, Man, and Cybernetics 24 (1994) 1374–1382.
[32] O. Stock (Ed.), Spatial and Temporal Reasoning, Kluwer Academic Publishers, 1997.
[33] Y. Shoham, Temporal logics in artificial intelligence: Semantical and ontological considerations, Artificial Intelligence 33 (1987) 89–104.
[34] D. Gabbay, I. Hodkinson, M. Reynolds, Temporal Logic (Mathematical Foundations and Computational Aspects), Clarendon Press, Oxford, 1994.
[35] F.D. Anger, R.V. Rodriguez, H.W. Guesgen, J. Van Benthem, Space, time and computation trends and problems, Applied Intelligence 6 (1) (1996) 5–9.
[36] H. Barringer, M. Fisher, D. Gabbay, G. Gough (Eds.), Advances in Temporal Logic, in: Applied Logic Series, vol. 16, Kluwer Academic Publishers, Dordrecht, Boston, 2000.
[37] R. Morris, L. Khatib (Eds.), Proceedings of the First International Workshop on Temporal Representation and Reasoning, Florida Artificial Intelligence Research Society, Daytona Beach, FL, 1994.
[38] R. Morris, L. Khatib (Eds.), Proceedings of the Second International Workshop on Temporal Representation and Reasoning, Florida Artificial Intelligence Research Society, Melbourne, FL, 1995.
[39] R. Morris, L. Khatib (Eds.), Proceedings of the Fourth International Workshop on Temporal Representation and Reasoning, IEEE Computer Society Press, Los Alamitos, CA, 1997.
[40] R. Morris, L. Khatib (Eds.), Proceedings of the Sixth International Workshop on Temporal Representation and Reasoning, IEEE Computer Society Press, Los Alamitos, CA, 1999.
[41] L. Chittaro, S. Goodwin, H. Hamilton, A. Montanari (Eds.), Proceedings of the Third International Workshop on Temporal Representation and Reasoning, IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.
[42] S. Goodwin, A. Trudel (Eds.), Proceedings of the Seventh International Workshop on Temporal Representation and Reasoning, IEEE Computer Society Press, Cape Breton, Canada, 2000.
[43] D. Gabbay, H. Ohlbach (Eds.), Proceedings of the First International Conference on Temporal Logic, Springer-Verlag, Bonn, Germany, 1994.
[44] D. Gabbay, H. Barringer (Eds.), Proceedings of the Second International Conference on Temporal Logic, in: Applied Logic Series, Kluwer, Manchester, UK, 1997.
[45] H.W. Guesgen, F.D. Anger, G. Ligozat, Proceedings of the Workshop on Spatial and Temporal Reasoning, Eighteenth IJCAI, 2003.
[46] A. Artale, E. Franconi, A survey of temporal extensions of description logics, Annals of Mathematics and Artificial Intelligence 30 (2000) 171–210.
[47] J.C. Augusto, The logical approach to temporal reasoning, Artificial Intelligence Review 16 (2001) 301–333.
[48] J. Ma, B. Knight, Reified temporal logics: An overview, Artificial Intelligence Review 15 (2001) 189–217.
[49] M.C. Golumbic, R. Shamir, Algorithms and complexity for reasoning about time, in: Proceedings of the 10th National Conference on Artificial Intelligence, AAAI, Press/MIT Press, 1992, pp. 741–747.
[50] M.C. Golumbic, R. Shamir, Complexity and algorithms for reasoning about time: A graph theoretic approach, Journal of the ACM 40 (5) (1993) 1108–1133.
[51] T. Drakengren, P. Jonsson, Eight maximal tractable subclasses of Allen's algebra with metric time, Journal of Artificial Intelligence Research 7 (1997) 25–45.
[52] I. Cervesato, M. Franceschet, A. Montanari, The complexity of model checking in modal event calculi, in: Proceedings of the Fourteenth International Conference on Logic Programming — ICLP'97, 1997.
[53] I. Cervesato, M. Franceschet, A. Montanari, A hierarchy of modal event calculi: Expressiveness and complexity, in: Proceedings of the Second International Conference on Temporal Logic, ICTL'97, 1997, pp. 1–17.
[54] I. Cervesato, M. Franceschet, A. Montanari, The complexity of model checking in modal event calculi with quantifiers, in: Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning — KR'98, 1998.
[55] A. Krokhin, P. Jeavons, P. Jonsson, Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra, Journal of the ACM 50 (5) (2003) 591–640.
[56] D. McDermott, A temporal logic for reasoning about processes and plans, Cognitive Science 6 (1982) 101–155.
[57] J. Allen, H. Kautz, R. Pelavin, J. Tenenberg, Reasoning About Plans, Morgan Kaufmann, San Mateo, CA, 1991.
[58] J. Allen, Temporal reasoning and planning, in: J. Allen, H. Kautz, R. Pelavin, J. Tenenberg (Eds.), Reasoning About Plans, Morgan Kaufmann, San Mateo, CA, 1991, pp. 1–68.
[59] J. Allen, Planning as temporal reasoning, in: Proceedings of KR '91, 1991, pp. 3–14.
[60] J. Allen, Formal models of planning, in: Readings in Planning, Morgan Kaufmann, 1991, pp. 50–54.
[61] C. Ma, On planning time sensitive operations, MS Thesis, SE, George Mason University, VA, 1999.
[62] M.E. Pollack, J.F. Horty, There's more to life than making plans: Plan management in dynamic, multi-agent environments, AI Magazine 20 (4) (1999) 71–84.
[63] N. Onder, M.E. Pollack, Conditional, probabilistic planning: A unifying algorithm and effective search control mechanisms, in: Proceedings of the 16th National Conference on Artificial Intelligence, AAAI-99, 1999.
[64] I. Tsamardinos, M.E. Pollack, J.F. Horty, Merging plans with quantitative temporal constraints, temporally extended actions, and conditional branches, in: Proceedings of the 5th International Conference on AI Planning Systems, Breckenridge, CO, 2000.
[65] J. Moder, C. Philips, Project Management with CPM and PERT, second edn, Van Nostrand Reinhold, NY, 1970.
[66] A.K. Zaidi, On temporal logic programming using Petri nets, IEEE Transactions on SMC, Part A 28 (1999) 3.

[67] C.L. Hamblin, Instants and intervals, in: F. Haber, J. Fraser, G. Muller (Eds.), in: The Study of Time, Springer-Verlag, New York, 1972, pp. 324–328.
[68] I. Meiri, Combining qualitative and quantitative constraints in temporal reasoning, Artificial Intelligence 87 (1996) 343–385.
[69] J.L. Peterson, Petri Net Theory and Modeling of Systems, Prentice Hall Inc., Englewood Cliffs, NJ, 1981.
[70] W. Reisig, Petri Nets, Springer-Verlag, NY, 1991.
[71] A.K. Zaidi, A.H. Levis, TEMPER: A temporal programmer for time-sensitive control of discrete-event systems, IEEE Transaction on Systems, Man, and Cybernetics 31 (6) (2001) 485–496.
[72] H.W. Guesgen, Fuzzifying spatial relations, in: P. Matsakis, L. Sztandera (Eds.), Applying Soft Computing in Defining Spatial Relations, Physica, Heidelberg, Germany, 2002, pp. 1–16.
[73] I. Rauf, A.K. Zaidi, A temporal programmer for revising temporal models of discrete-event systems, in: Proc. of 2002 IEEE International Conference on Systems, Man, and Cybernetics, Hemmamat, Tunisia, 2002.
[74] A. Gerevini, L. Schubert, Efficient algorithms for qualitative reasoning about time, Artificial Intelligence 74 (2) (1995) 207–248.
[75] A. Galton, Logic for Information Technology, John Wiley & Sons, 1990.
[76] J. Martinez, M. Silva, A simple and fast algorithm to obtain all invariants of a generalized Petri net, in: Informatik-Fachbrichte, vol. 52, Springer-Verlag, 1982, pp. 301–310.
[77] G. Memmi, J. Vautherin, Analysing Nets by the Invariant Method, in: Lecture Notes in Computer Science, vol. 254, Springer, Berlin, Heidelberg, New York, 1987, pp. 300–336.
[78] R.G. Busacker, Finite Graphs and Networks: An Introduction with Applications, McGraw-Hill, New York, 1965.
[79] S. Warshall, A theorem on boolean matrices, Journal of the ACM 9 (1) (1962) 11–12.
[80] H.A. Taha, Operations Research: An Introduction, McMillan Publishing Company, 1992.

Project Management Using Point Graphs

Mashhood Ishaque

Abbas K. Zaidi

Alexander H. Levis

Point of Contact: **Mashhood Ishaque**
Address: 161 College Ave,
Medford, MA, 02155
Phone: 703 945 3601
Email: mishaq01@cs.tufts.edu

• *The work was carried out with support provided by the Air Force Office of Scientific Research under contract numbers FA9550-05-1-0106.*

System Architectures Laboratory

Project Management Using Point Graphs[1]

Abstract

A graph-based approach for scheduling and monitoring temporal events and activities in a system engineering project is presented. It attempts to overcome some of the limitations of traditional project management approaches by allowing specifications of real-time *milestones*, and by breaking the *finish-start* barrier between the activities. An example of a residential construction project is used to illustrate the approach.

1. Introduction

The traditional project management and concurrent engineering approaches date back to 1950's with the advent of the Critical Path Method (CPM), which was developed as a result of a joint venture between DuPont Corporation and Remington Rand Corporation to study the trade-offs between the cost of a project and its overall completion time. In 1958, another approach called Project Evaluation and Review Technique (PERT) was invented by Booz Allen Hamilton, Inc. under contract to the US Department of Defence's US Navy Special Projects Office as part of the Polaris missile project with an emphasis on shortening the overall completion time of the system under development.(Moder and Philips, 1970) Elmaghraby in his invited review (1995) reports that the first paper on the two approaches, commonly called CPM/PERT, appeared in 1959 (Kelly and Walker, 1959). These are essentially paper-and-pencil techniques that have been implemented as (or in several) software tools by many organizations and individuals, and have been extensively used by professionals since their inception. A good review of the literature on, and available software implementations of, CPM/PERT can be found in (Elmaghraby, 1995). For an account of more recent practices in the project management area, in general, and in CPM/PERT type techniques, in particular, readers may refer to PMBOK

[1] The work was carried out with support provided by the Air Force Office of Scientific Research under contract number FA9550-05-1-0106.

guide (2004), (Kis, 2005) which provides a review of some recent books on project scheduling, and to (Browning and Ramasesh, 2007).

The CPM/PERT techniques address a narrow aspect of a project management or a system engineering problem that is called 'Activity Time Management' or 'Time Analysis' of *project networks*. In this paper, we keep our discussion to this problem only and do not address other aspects, e.g., resource constraints/availability/types, financial issues, uncertainty in activity durations and/or resource availability, etc., in a more general project/product/system management/engineering setup.

The CPM/PERT are network (i.e., graph) based techniques that provide an analytic underpinning to product/project management/monitoring problems by addressing the shortest completion time, time-cost tradeoffs, and scheduling issues (i.e., critical activities and time slacks) for both deterministic and probabilistic concurrent activities. The network or graph representations employed by these approaches to model the project activities, and temporal constraints among them, also provide visual insights about the temporal aspects of a project plan. The network representations, called project networks and/or activity nets, come in two different flavours: Activities on Arcs (AOA) and Activities on Nodes (AON). Each representation has its own advantages and disadvantages in terms of modelling, visualization, and computational complexity. There are many research papers only on issues related to these two types of network representations: issues like uniqueness of a representation; complexity of a representation and of reduction techniques (Kamburowski et al., 2000); transformation from one representation to another (Cohen and Sadeh, 2007) are some example problems found in the research literature.

The classical network based approaches (regardless of the network representation used) have certain limitation in terms of the type of activities and the types of temporal constraints allowed between activities. For example, they are limited to only duration based activities, and the only temporal constraint allowed is between the end of an activity and the start of a following activity, referred to as the finish-start barrier in this paper (which is also referred as end-to-start precedence constraint in literature). This

precedence constraint between two activities, say h and k, implies that activity h must be completed before activity k can be initiated. This results in the specification of sequential and totally parallel activities, with no provision of requirements for *nominally* parallel tasks/activities, i.e., activities with more general temporal constraints among the starts, and ends of activities with possible lead/lag durations/delays. These temporal relations are called 'Generalized Precedence Relations (GPRs)' and are specified as constraints between pairs of activities (Elmaghraby and Kamburowski, 1992). The GPRs are translated into four specific types of temporal constraints (with the assumption that an activity is indivisible and can be modelled only with 'start' and 'finish' events): the start-to-start (SS), the finish-to-finish (FF), the start-to-finish (SF), and the finish-to-start (FS) relations. (Elmaghraby and Kamburowski, 1992; Demeulemeester and Herroelen, 2002; Dorndorf, 2002; PMBOK guide, 2004)

Elmaghraby and Kamburowski (1992) claimed to have provided the first comprehensive treatment of GPRs for project networks and CPM type analysis. They also provided a historical perspective of their approach and an account of earlier (partial) attempts on the problem. Their approach is based on an earlier formalism by Kerbosch and Schell (1975) that represents GPRs as a set of linear inequalities comprising of algebraic expressions of end points of activities and lead/lag values. The approach in (Elmaghraby and Kamburowski, 1992) first converts all the temporal constraints into this analytical representation, then constructs a graphical representation unlike any of the two AON or AOA representations, and then applies a combination of a shortest path tree algorithm and a mathematical program to perform the temporal analysis of the project. The graphical representation used in this approach is identical to the *distance graphs* (proposed around the same timeframe) by Dechter et al. (1991) that are used to model Simple Temporal Problem (STP). Unlike classical CPM, this approach is not a paper-and-pencil technique and requires computer assistance. The graphical representation of the distance graphs is also not as intuitive as that of CPM networks. In fact, the illustrative example presented in (Elmaghraby and Kamburowski, 1992) has errors both in the set of inequalities representing a given set of temporal constraints among activities,

and in the corresponding distance graph constructed from these inequalities that went unnoticed, perhaps, for the same reasons.

The size, complexity, and distributed nature of modern-day systems (or systems of systems) and the engineering processes required to design such systems call for a robust set of tools supporting different aspects of the system engineering and management processes. The approach presented in this paper contributes to this tool suite by overcoming some of the weaknesses in the traditional approaches to project management and monitoring.

This paper proposes a new graph-based representation, called Point Graphs (PG), to model a class of temporal relations between points (instantaneous activities) and/or intervals (activities with duration). The points and intervals represent time stamps and time delays (durations) associated with events and activities in a project plan, respectively. The graph representation is shown to enable analyses similar to that of classical CPM for projects with GPRs among activities. The graph representation is supported by an input specification language (i.e., a structured natural language) for project activities and temporal constraints among them. The language and the graph representation used in the approach are more expressive than the project networks used for the traditional approaches. The graph structure of the PG is also used to reveal and identify corrections for inconsistencies and temporal anomalies, if present in the input.

The Point Graphs (PG) and the accompanying *Point-Interval* formalism (PIL) originated from an earlier work (Zaidi, 1999, 2000; Zaidi and Levis, 2001; Ishaque and Zaidi, 2005; Zaidi and Wagenhals, 2006) on temporal knowledge representation and reasoning. An attempt is made in (Zaidi and Wagenhals, 2006) to use the PG representation for project networks and PIL formalism for solving time activity management problems with GPRs. The resulting approach used a combination of graph algorithms and a mathematical program for the solution, i.e., to determine the shortest duration for the entire set of activities, earliest/latest start/end times of temporal activities, critical activities, and time slacks for non-critical activities. The approach in (Zaidi and Wagenhals, 2006) is extended in this paper by:

1. . Adding the provision of lead and lad times/durations with GPRs among activities. For example, a temporal constraint of the type “The start of Activity X is no earlier than 15 hours and no later than 10 hours, *before* the end of Activity Y” can be handled by the new approach.
2. Allowing the provision for exact deadlines, i.e., milestones, for start/end of activities, specification of ‘at least’ and ‘at most’ type constraints on start/end/completion times of activities, etc. This feature is especially useful for a real-time monitoring of a project’s progress with activities either falling behind the schedule and/or being completed earlier than scheduled. The results of the analyses can help project managers re-adjust their project schedules/plans with real-time feedback.
3. Solving the time management problem for a given set of activities and temporal constraints among them using only the PG representation and an extended suite of graph algorithms. The new approach does not require the mathematical program formulation.
4. Implementing the approach in a software application that incorporates a hierarchical Point Graph representation to support both top-down and bottom-up project management paradigms, which is especially useful for large-scale systems.

The paper is organized as follows: we describe the Point Graph representation in Section 2. This section has been added in an attempt to make this paper self-contained; similar, even more detailed, description of PGs can be found in earlier papers. The PG based Project Scheduling algorithms are presented in Section 3, and Hierarchical Point Graphs in Section 4. In Section 5 we give an illustrative example for the approach, and in Section 6 we present Temper—the software implementation of the approach.

2. Point Graphs

Figure 1 presents a graph construct called Point Graphs (PG) that is used to model temporal information in a project. In a PG, a node represents a point (or a *composite* point) and an edge between two points represents one of the two temporal relations,

before (<) and *precedes* (≤), between the two. An interval, i.e., X = [sX, eX], in this graph representation is depicted as a pair of start and end points with an edge connecting the two.

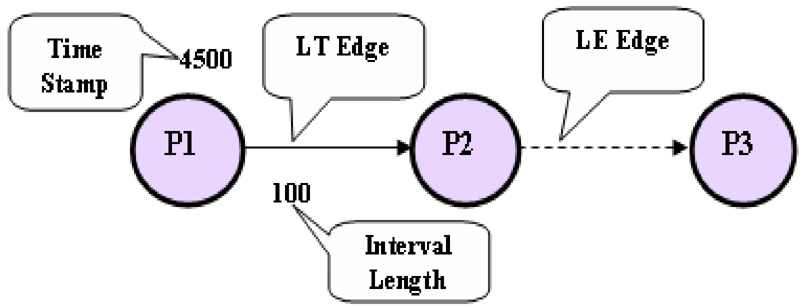


Figure 1. Point Graph Construct.

Definition 1:

A Point Graph, PG (V, E_A, D, T) is a directed graph with:

V: Set of vertices with each node or vertex v ∈ □ V representing pointinstant on the time line. Points Pi, Pj, …, Pn are represented as a composite point [Pi; Pj; …; Pn] if all are mapped to a single point on the line.

E_A: Union of two sets of edges: $E_A = E \cup E_{\leq}$, where

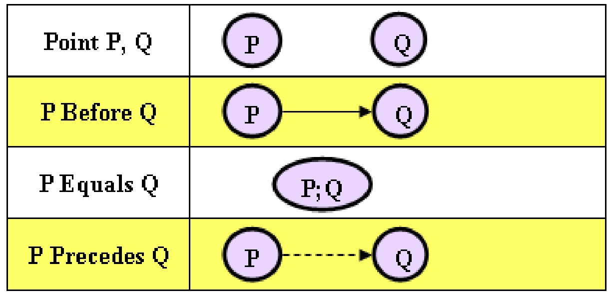
E : Set of edges with each edge e_{12} ∈ □ E, between two vertices v1and v2, also denoted as (v1, v2), representing a relation '<' (*before*) between the two vertices - (v1 < v2). The edges in this set are called LT edges;

$E_{\leq}$: Set of edges with each edge $e_{12} \in E_{\leq}$, between two vertices v1 and v2, also denoted as (v1, v2), representing a relation '≤' (*precedes*) between the two vertices - (v1 ≤ v2). The edges in this set are called LE edges.

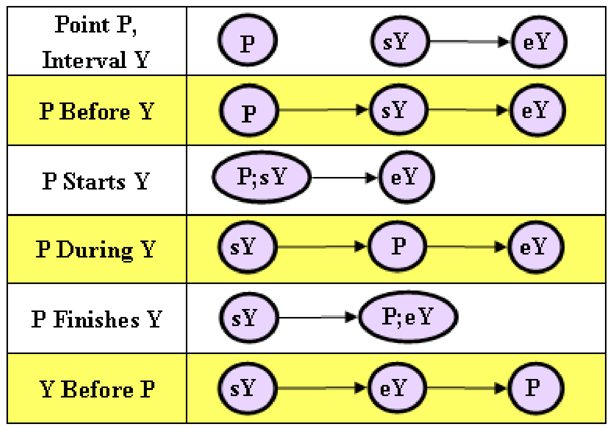
D: Edge-length function (every edge is assigned a length): E ∈ $\Re$□

T: Vertex-stamp function (a vertex may or may not have stamp): V ∈ $\Re$

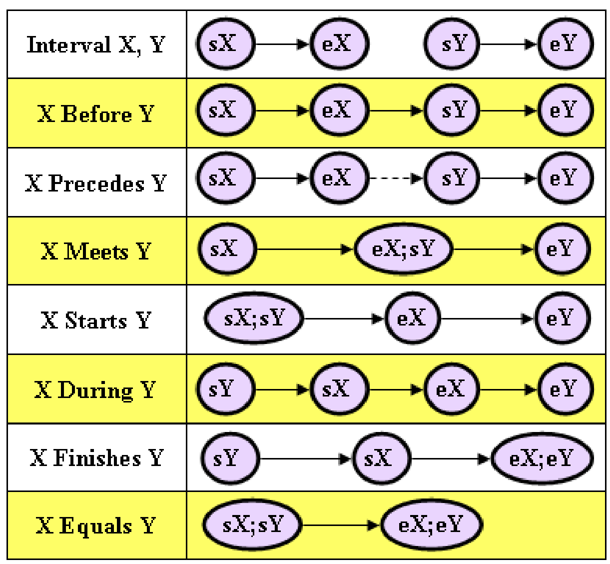
Figures 2 and 3 illustrate how this PG representation can be related to constructs of an input language. For a more detailed description of this language, see (Zaidi and Wagenhals, 2006). The language, called PIL, can be used to specify temporal constraints,



a) Point to Point (PP) Constraints



b) Point to Interval (PI) Constraints



c) Interval to Interval (II) Constraints

Figure 2. Specification Language and PG Representation.

both qualitative and quantitative, among intervals and events representing tasks and milestones in a project. The temporal constraints in Figure 3 require introduction of

virtual time point(s) or virtual nodes in a PG. A virtual node is like any other node in a PG except for the fact that there is no temporal variable (point, start of interval, or end of interval) associated with it. It, therefore, does not have a unique identifier or *name* associated with it.

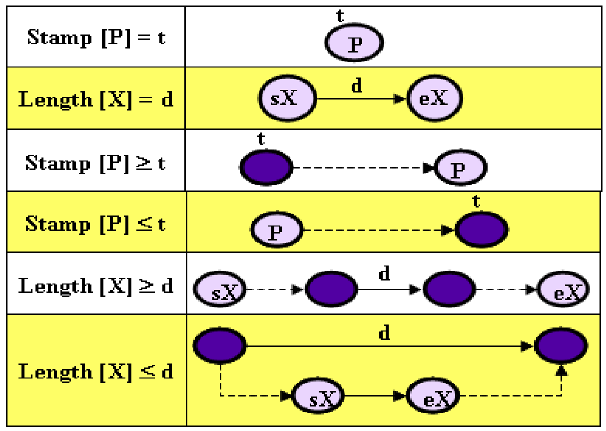


Figure 3. Quantitative Constraints and their PG Representations.

Operations on Point Graphs

We start with a set of PIL statements representing the temporal entities and the constraints to be satisfied. Each statement in this set can be converted to an equivalent graphical representation using the mapping shown in Figures 2-3. The PG representing the entire set of statements is then constructed by *unifying* individual PGs to a (possibly) single connected graph. The unifying process looks at the labels of the nodes (except for virtual nodes) and the values of the stamps associated with them to identify equalities. The nodes identified as being equal to one another are merged into a single node with a composite label. The unified PG is then *folded* with the help of lengths on edges. This folding process establishes new relations among system intervals, inferred through the quantitative analysis of the known relations specified by interval lengths and stamps.

Figure 4 illustrates the two operations on an example set of PGs. The input constraints may be infeasible; therefore, the resulting PG is checked for inconsistency, as illustrated in Figure 4c. The infeasibility in the input constraints reveals itself in a PG either in the form of cycles or with presence of multiple paths between a pair of nodes

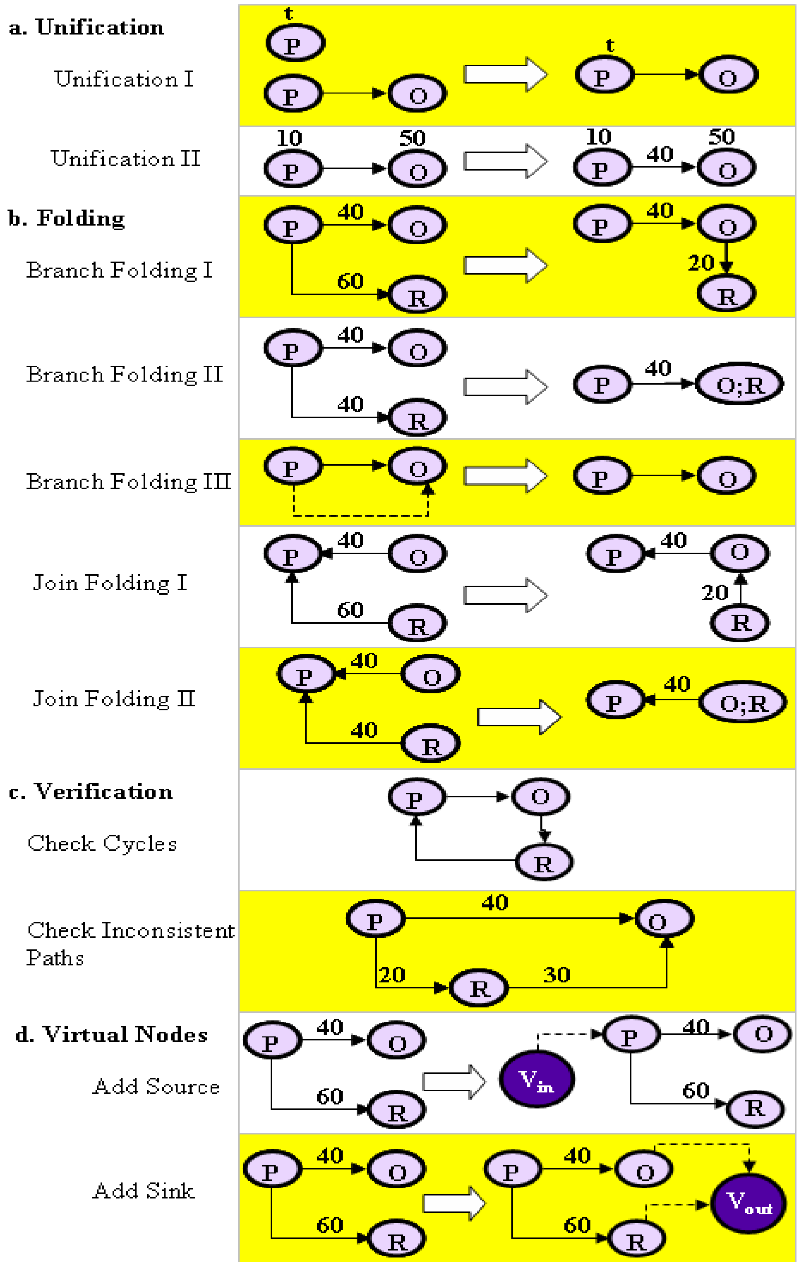


Figure 4. Operations on a Point Graph.

with conflicting path lengths. A more technical and detailed description of these graph operations, their computational complexity, and the verification mechanism can be found in Zaidi and Wagnehals (2006), and Ishaque (2006). Once a PG is verified, a couple of virtual nodes are added to it as illustrated in Figure 4d. A virtual source (sink) node V_{in}

(V_{out}) is added to PG with LE arcs connecting V_{in} (V_{out}) to all the source (sink) nodes in the PG as shown in the figure. The two nodes represent the overall start-to-end time for the project under consideration. Once a PG for the project is constructed with the help of steps illustrated in Figure 4, a set of algorithms, presented in the following section, are applied to it.

Example 1.

The following is an example set of PIL statements representing activities and temporal constraints for a fictitious project:

Interval A, B, C, D, E

Length A = 5	Length B = 5	Length C = 5	Length D = 2
Length E = 10	A Meets B	C Meets D	C Precedes B

eE Precedes eD

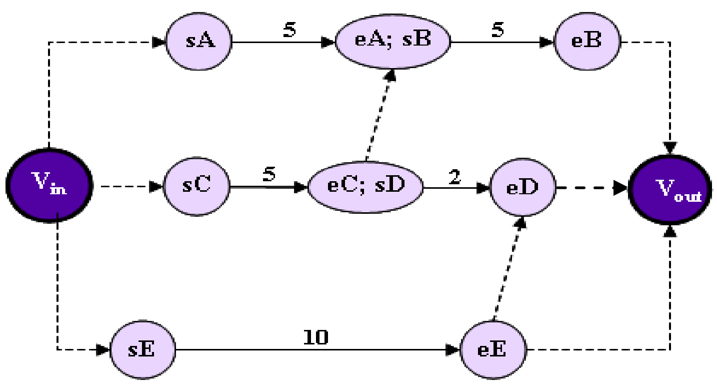


Figure 5. PG for Example 1.

In the example set, some of the constraints, namely 'A meets B', 'C meets D' and 'eE Precedes eD', are the type of constraints that cannot be modelled in the traditional approaches. A quick comparison between the PG representation and the two types of (i.e., Activity on Arc and Activity on Node) project networks used by CPM/PERT (Moder and Philips, 1970) is presented in Figure 6. The last three PGs have no equivalent representations in the project networks.

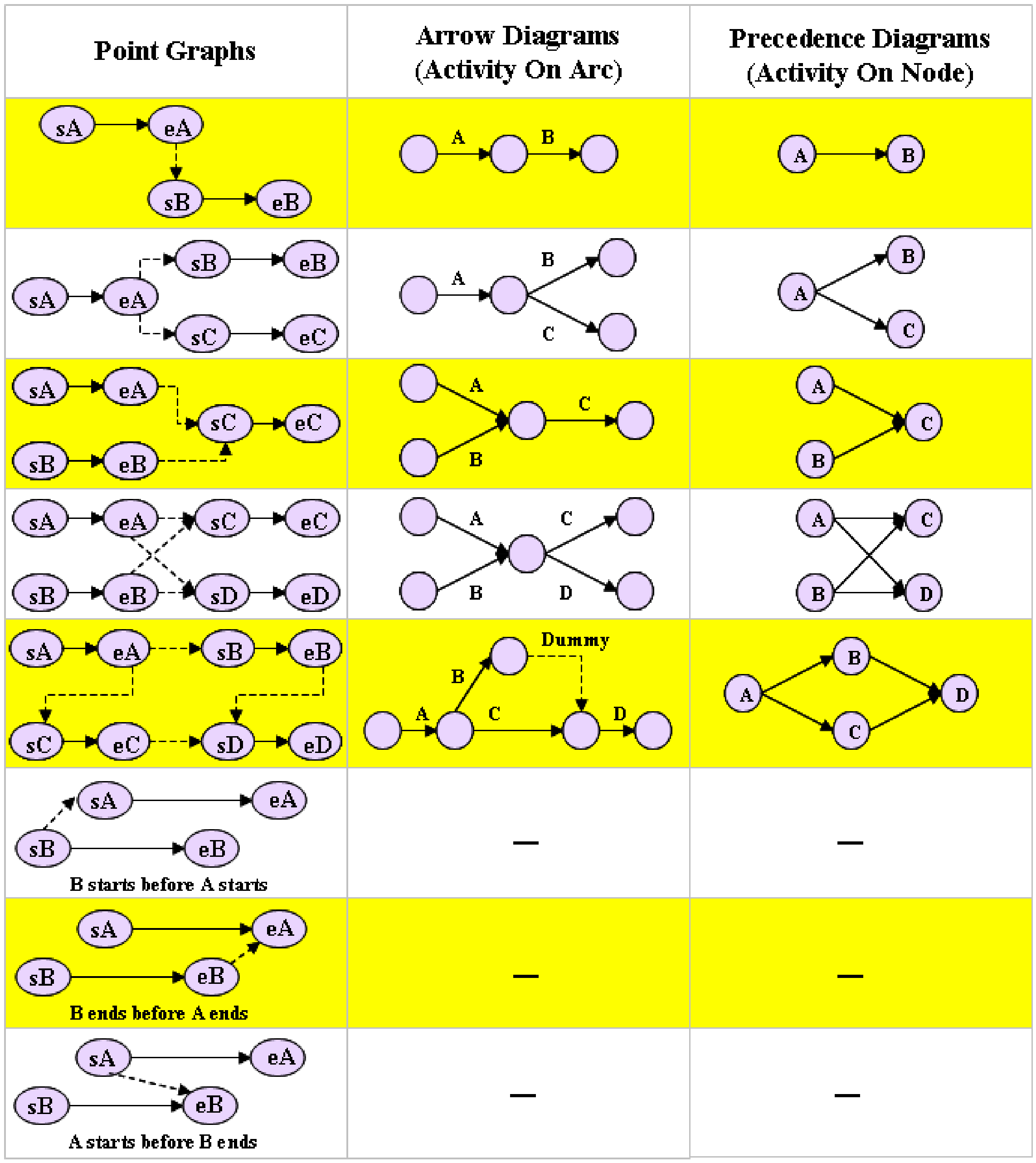


Figure 6. A Comparison between PG Representation and Project Networks.

3. Scheduling Algorithms

The scheduling algorithms applied to the PG representation calculate three parameters for each node in the PG. The parameter values are calculated by running two sets of

algorithms, *Forward** followed by *Reverse**, on the graph. The values of these parameters help determine the *critical activities* and time floats/slacks for intervals in the system, and *interval/point activities* defined for the PG under consideration. The three parameters are called *earliest occurrence (Ev), late occurrence (Lv), and latest occurrence (Tv)* of a node ‘v’, and are defined as follows.

Earliest Occurrence, E_v of a node v is the smallest time stamp on the node that satisfies the earliest occurrences of the preceding nodes. Figure 7 illustrates the method of calculating a node’s earliest occurrence time with the help of earliest times on its child (preceding) nodes. The manner in which this parameter is calculated for all the nodes in a PG requires a *forward* traversal of the PG starting from the sink node, which by default is given a 0 value for the earliest occurrence time.

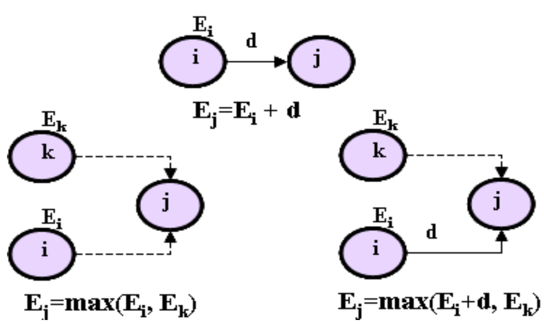


Figure 7. Earliest Occurrence Time.

Late (Latest) Occurrence, L_v (T_v) of a node v is the largest time stamp on the node that satisfies the earliest (latest) occurrences of the following nodes. Figure 8 illustrates the method of calculating a node’s late and latest occurrence times with the help of corresponding parameter values on its parent (following) nodes. The calculation for these two parameters requires a *reverse* traversal of the PG starting from the sink node, which is by default initialized to the earliest occurrence time, calculated during the forward sweep, for both late and latest occurrence times.

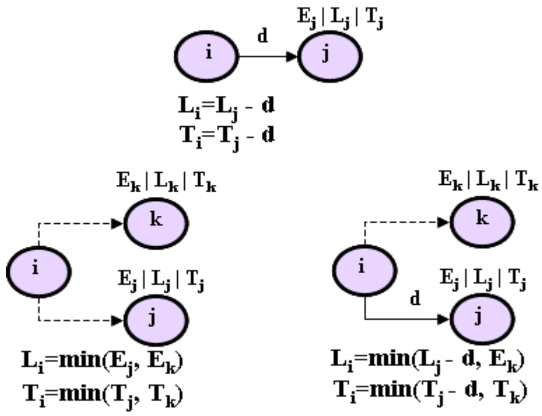


Figure 8. Late & Latest Occurrence Times.

Notice when a stamp (i.e., as a deadline or milestone) has been provided for a node in the Point Graph, the stamp represents a hard constraint on the occurrence time of the node. Thus the earliest, late and latest occurrence times must be equal to the stamp of the node and; the calculation as illustrated in Figures 7 and 8 are adjusted accordingly. The two passes on the PG representation are now presented in more detail in the following sections.

Forward*

The forward pass comprises two algorithms, as described in Tables 1 & 2, which calculate E_v for each node v in the PG. The first algorithm, the Forward I (Table 1) simply traverses the graph and calculates the parameter values as illustrated in Figure 7. The time complexity of the algorithm is O(m+n) where m is the number of edges and n is the number of nodes in a PG. After the application of this algorithm, it is possible for an activity to have *Stretch Float*, i.e., the earliest start of the activity does not ensure its earliest completion; or the activity needs to be stretched beyond its duration in order to complete the project on time. The concept was first introduced in (Zaidi and Wagenhals, 2006.) A real life example of stretch float can be found in air transportation. A plane takes off at location A and lands at location B, the time difference between the landing and take off may be greater than the minimum travel time between the two destinations.

In case an activity is not stretchable, the earliest occurrence time needs to be recalculated in the PG. Figure 9 shows the PG in Example 1 (Figure 5) after it was processed by the Forward I algorithm. Activity D is shown with a stretch float equal to 3.

Table 1. The Forward I Algorithm.

Calculating E_v for node v
(starting from V_{out})
Let *v be the pre-set of v
Check for node stamp:
IF v has stamp
Set E_v = stamp[v]
ELSE
Set $E_v = 0$ //initialization
FOR each node v_i in *v
Calculate E_{vi} recursively.
IF v does not have stamp
IF (v_i,v) is LT edge
$E_v = \max(E_v, E_{vi} + \text{Length}(v_i, v))$
ELSE If (v_i,v) is LE edge
$E_v = \max(E_v, E_{vi})$

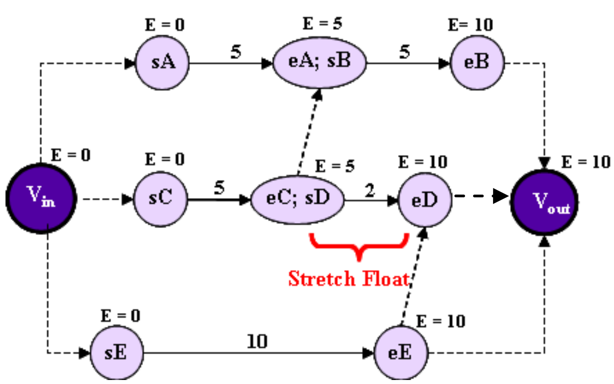


Figure 9. A PG with Stretch Float

The second algorithm, called Forward II (Table 2) re-calculates the earliest occurrence time for each activity without the stretch float. The algorithm is not needed if the application of Forward I results in no reported stretch floats or if the reported stretch floats are associated with stretchable activities. The algorithm, when applied, works by

fixing the edges where stretch float occurs. Fixing an edge might result in a situation where other edges (both LT and LE) may need to be fixed as well. But the algorithm always fixes at least one edge in each iteration, so a maximum of |E| iterations will fix all edges. In each iteration, the algorithm examines all edges. So the time complexity of the algorithm is $O(m^2)$. The two algorithms are collectively called Forward*.

Table 2. The Forward II Algorithm.

```
FOR |E| times
(where E is set of edges in the PG)
    Set Modified = false.
    FOR each edge (v_i, v_j)
        IF (v_i, v_j) is a '≤' edge
                IF E_i > E_j
                Set E_j = E_i. Set Modified = true.
        ELSE
            IF E_j > E_i + Length(v_i, v_j)
             Set E_i = E_j - Length(v_i, v_j).
             Set Modified = true.
            ELSE IF E_j < E_i + Length(v_i, v_j)
                Set E_j = E_i + Length(v_i, v_j).
                Set Modified = true.
    IF (Modified = false)
    Return
```

Correctness of Forward II Algorithm

Here we give a sketch of the proof of correctness for Forward II algorithm. In Figure 10 we give an example of a Point Graph with inconsistent paths for which the algorithm will not work. Next we show that the algorithm is correct for consistent Point Graphs.

Notice that a consistent Point Graph can be mapped on the timeline such that there is no stretch float. Further observe that given such a mapping, we can always shift (each connected component of) the Point Graph on the timeline such that at least one activity starts at time instant '0', provided there are no hard constraints (in the form of

time stamps). We are interested in such a mapping where all the activities start at their earliest (with no stretch float).

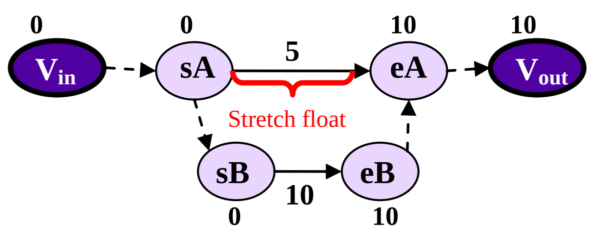


Figure 10. A PG with Stretch Float

The algorithm works by pushing activities towards the right. In every iteration at least one activity is pushed towards the right. But this cannot continue forever because the maximum project completion time is upper bounded by the sum of durations for all activities and there is an activity in (each connected component of) the Point Graph which cannot be moved towards the right.

Reverse*

The application of Forward* is followed by a similar set of algorithms, collectively called Reverse*, for the calculation of late and latest occurrence times for every node in the PG. Tables 3 and 4 describe the two algorithms for late occurrence time. The calculation of latest follows the same steps as in Tables 3-4 with the only difference in the formula used to calculate the parameter values. A proof of correctness for Reverse* follows the same line of arguments as presented for Forward*. Figure 11 shows the PG of Example 1 after it is processed by both Forward* and Reverse* algorithms with no provision for stretch float which required the application of the Forward II and Reverse II algorithms on it.

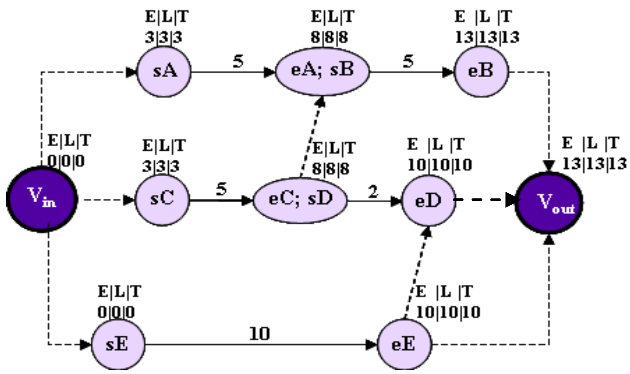


Figure 11. PG of Example 1

Table 3. The Reverse I Algorithm.

Calculate L_v for node v (starting from V_{in})
Let v* be the post-set of v
Set $L_v = E_v$ //initialization
FOR each node vi in v*
  Calculate L_{vi} recursively
  IF v does not have stamp
    IF (v,vi) is LT edge
      $L_v = \min(L_v, L_{vi} - \text{Length}(v, v_i))$
    ELSE IF (v,v_i) is '≤' edge
      $L_v = \min(L_v, E_{vi})$

Table 4. The Reverse II Algorithm.

FOR |E| times (where E is the set of edges in PG)
  Set Modified = false.
  FOR each edge (v_i, v_j)
    IF (v_i, v_j) is a '<' edge
      IF $L_j > L_i + D(vi, vj)$
      Set $L_j = L_i + D(v_i, v_j)$.
      Set Modified = true.
    ELSE IF $L_j < L_i + D(v_i, v_j)$
      Set $L_i = L_j - D(v_i, v_j)$.
      Set Modified = true.
  IF (Modified = false)
  Return

Finally the PG corresponding to a project's requirements with the values of the parameters calculated can be used to construct a time chart, e.g., a Gantt chart, showing the start and finish times for each activity as well as its relationship to other activities. The parameters values are used to identify *critical* and non-critical activities in the project. For non-critical activities, the amounts of slacks or *floats* are calculated so that they can be used advantageously when such activities are delayed or when limited resources are to be used. The following describes the notion of a critical activity and different types of floats that are calculated after the application of graph algorithms. Table 5 shows these calculated values for the system in Example 1.

Critical Activity. An activity is defined to be critical if a delay in its start will cause a delay in the completion time of the entire project, i.e.,

i. For a point activity $v \in V$, $Ev = Tv$; for an interval activity [v1, v2], where v1, v2 $\in V$, $v \in \{v1, v2\}$, $Ev = Tv$, *or*

ii. For an interval activity, it 'Meets' or is met by another critical activity; for a point activity, it 'Starts' and/or 'Ends' another critical activity.

Total Float (TF) and Free Float (FF). Total Float (TF) is the difference between the maximum time available to perform an activity and its duration. Free Float (FF) for an activity is defined by assuming that all the activities start as early as possible; it is the excess time available over its duration.

(a) Total float (TF) and free float (FF) for a non-critical point activity v are calculated as:

$TFv = Tv - Ev$

$FFv = Lv - Ev$

(b) Total float (TF) and free float (FF) for a non-critical interval activity [v1, v2] are calculated as:

$TF_{[v1, v2]} = Tv2 – Ev2 = Tv1 – Ev1$

$FF_{[v1, v2]} = Lv2 – Ev2 = Lv1 – Ev1$

(For all critical activities TF = FF = 0.)

Table 5. Parameter Values for Activities in Example 1.

Activity	Duration	Earliest Start Time	Latest End Time	Critical	Total Float	Free Float
A	5	3	8	Yes	0	0
B	5	8	13	Yes	0	0
C	5	3	8	Yes	0	0
D	2	8	10	Yes	0	0
E	10	0	10	Yes	0	0

Project Monitoring

Since all the algorithms presented in the paper have polynomial time complexity, their software implementation not only can handle large scale system projects but can also be used for real-time monitoring of project plans. The allowance for exact stamps or for specifying bounds on stamps for start/end of activities in the approach can be used to add additional constraints (especially during project execution) reflecting actual and more accurate times/dates for the activities involved. With the new information added, better estimates for completion times and time slacks can be re-calculated to determine if things are falling behind the schedule and/or being completed earlier than scheduled. The results of the analyses can help project managers re-adjust their project schedules/plans/cost estimates with real-time feedback. The following example illustrates the approach.

Example 2.

Suppose in the system of Example 1, the new information requires that Activity E can start no earlier than time 3 (Stamp [sE] ≥ 3), and Activity B must start at time 12 (Stamp [sB] = 12).

Figure 12 shows the new PG constructed for the new input with the new parameter values calculated for each node. The figure does not show the virtual nodes and is a simplified version of the actual PG that will be generated for the input. Table 6 shows the new parameter values for the activities involved in the project.

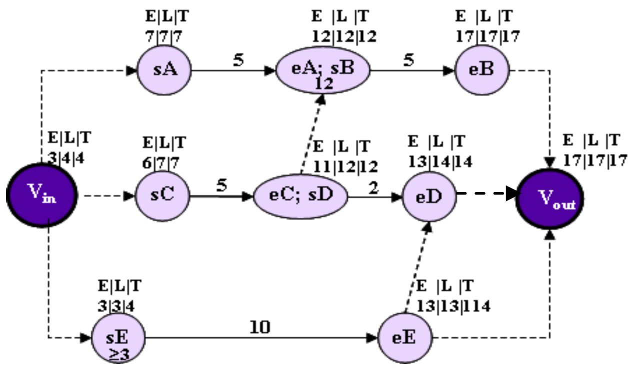


Figure 12. PG for Example 2.

Table 6. Parameter Values for Activities in Example 2.

Activity	Duration	Earliest Start Time	Latest End Time	Critical	Total Float	Free Float
A	5	7	12	yes	0	0
B	5	12	17	yes	0	0
C	5	6	12	No	1	1
D	2	11	14	No	1	1
E	10	3	14	No	1	0

A linear program formulation of the scheduling problem that minimizes the end-to-end time (V_{out} – V_{in}) and translates the PG representation to analytical inequalities representing the temporal constraints has been presented in (Zaidi and Wagenhals, 2006). The linear program formulation however lacks the advantages of the graphical representation and the inference mechanism supported by the PG representation.

4. Hierarchical Point Graph

A Hierarchical Point Graph (HPG) facilitates hierarchical planning by enabling a manager to work at a higher level of abstraction but still generate feasible plans consistent with the low-level system details. Hierarchical point graphs provide the capability to combine the high-level plan with the detailed sub-plans. Figure 13 illustrates this hierarchical arrangement with the help of the PG from Example 2 in which the activity E is shown substituted by a detailed PG with the low-level activities and constraints.

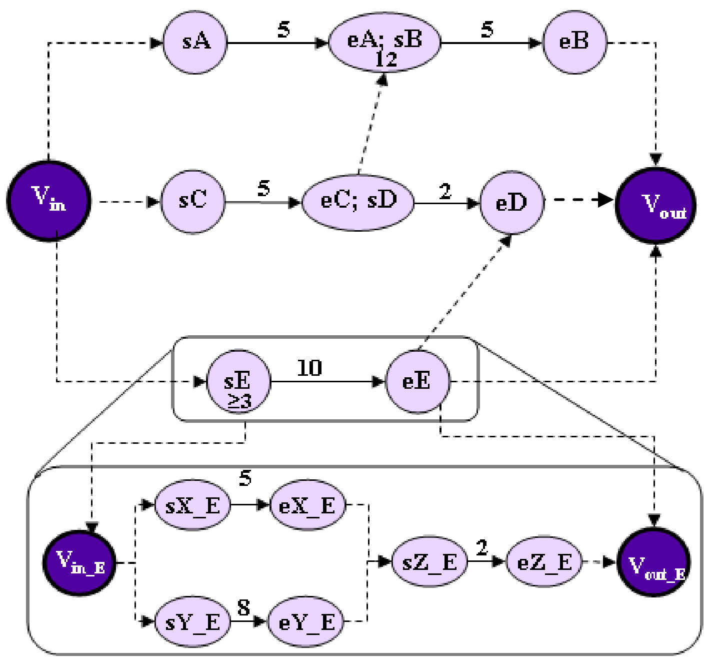


Figure 13. A Hierarchical PG

Definition 2:

A Hierarchical Point Graph, HPG (PG, M) is a directed graph with:

PG: A Point Graph, PG (V, E_A, D, T) (defined earlier)

M: A set of ordered pairs; it maps a pair of node in PG to another Hierarchical Point Graph;

$M = \{((a, b), HPG_{ab}) \mid a, b \in V$ and there is no directed path from b to a in PG, and HPG_{ab} is a Hierarchical Point Graph$\}$

Notice that for ordinary Point Graphs, the set M is empty. Since M is a relation, it is possible to associate multiple Hierarchical Point Graphs with a single interval (pair of nodes), which corresponds to the case when an activity in the high-level Point Graph can be substituted by multiple low-level parallel activities. Hierarchical Point Graphs enable planning at arbitray levels of increasing detail. An interval in a Point Graph can be subsituted by yet another Point Graph. This process of subsitution can continue unitl the intervals in the Point Graph represent a system's primitive actions.

HPGs encourage a modular and distributive approach to project management in which detailed plans can be developed separately from the high-level tasks representing them with completion times used as constraints that bind different levels together. This hierarchical manner of constructing project plans can be done using both bottom-up and top-down approaches. In the bottom up approach, completion times for sub-plans can be added as constraints for the abstract high-level activities representing them. In the top-down approach, the earliest start and latest completion times calculated for a high-level activity become constraints for the low-level sub-plan substituting the activity.

HPGs also enable plan re-use, where an organization has a set of well-defined high-level tasks and maintains a repository of sub-plans corresponding to each of these high-level tasks. There may be multiple sub-plans to choose from for a high-level task, with differing performance and cost characteristics. These sub-plans are re-used to quickly assemble a high-level plan. This re-use of already built (and tested) plans may result in the overall reduction of plan development time and may improve the reliability of the generated plan.

5. Illustrative Example – Residential Construction

We use the modified version of project template "Residential Construction Project Plan" for Microsoft Office Project 2007 [MS Project] to illustrate the scheduling approach presented in this paper. The project plan in the example is for the construction of a single family home. A set

of high-level activities for the project are given in Table 7, along with their estimated durations in days. The table also shows a point activity B ("Apply for permits"). The activities in Table 7, along with the temporal relationships (Table 8) among them, are shown in the form of a Point Graph in Figure 14. The two virtual nodes V_{in} and V_{out} have been shown added to the Point Graph in the figure marking the start and completion of the project, respectively.

Table 7: Activities in the Project

Activity	Description	Estimated Duration (Days)
A	Finalize plans, estimates, and sign contract	21
B	Apply for permits (point activity)	0
C	Lay foundations and basement walls	42
D	Complete framing and dry-in	44
E	Finish Exterior	19
F	Finish Interior	30
G	Finish landscaping and groundwork	13
H	Final inspection	8

Table 8: Temporal Relationships Among Activities

Natural Language Description	Temporal Relationship
Finalize plans and estimates before applying for permits	A Precedes B
Apply for permits before laying foundations	B Precedes C
Lay foundation before starting framing (and dry-in)	C Precedes D
Framing and dry-in must be complete before finishing can start	D Precedes E D Precedes F
Exterior finishing must be complete before landscaping can start	E Precedes G
Landscaping and finishing should be complete before the house is ready for inspection	F Precedes H G Precedes H

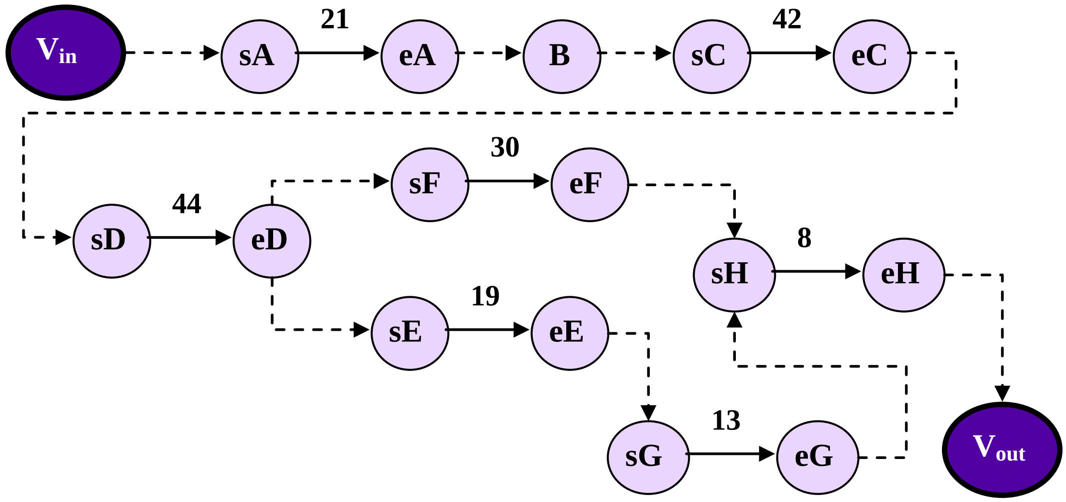


Figure 14. High-level Point Graph for Residential Construction Project

The attributes for the activities in the project are calculated by running the scheduling algorithms on the Point Graph as shown in Table 9. Observe that all activities are critical except F, which has a total and a free float of 2 days. The minimum completion time (makespan) of the project is 147 days (calculated by subtracting the latest end time of V_{in} from the latest end time of V_{out}).

Table 9: Attributes for the Activities in the Project

Activity	Duration (Days)	Earliest Start Time	Latest End Time	Critical	Total Float	Free Float
V_{in}	0	0	0	yes	0	0
A	21	0	21	yes	0	0
B	0	21	21	yes	0	0
C	42	21	63	yes	0	0
D	44	63	107	yes	0	0
E	19	107	126	yes	0	0
F	30	107	138	no	2	2
G	13	126	139	yes	0	0
H	8	139	147	yes	0	0
V_{out}	0	147	147	yes	0	0

Now suppose for the purpose of illustration that the activity D ("Complete framing and dry-in") is going to be performed by two different contractors, and we need a plan that schedules the sub-activities for the two contractors. The sub-activities for D and their durations are given in Table 10, the temporal constraints among them are given in Table 11, and the corresponding

Point Graph is shown in Figure 15. We will use activity D to illustrate both top-down and bottom-up approaches for project management with the PG representation.

Table 10: Sub-Activities for Framing and Dry-in

Activity	Description	Estimated Duration
1_D	Frame first floor decking	5
2_D	Frame first floor walls and corners	6
3_D	Frame second floor decking	5
4_D	Frame second floor walls and corners	6
5_D	Complete roof framing	9
6_D	Install first floor sheathing	7
7_D	Install second floor sheathing	7
8_D	Install roof decking	8

Table 11: Temporal Relationships Among Sub-Activities for Activity D

Natural Language Description	Temporal Relationship
First floor decking must finish before first floor wall and corners can be framed	1_D Precedes 2_D
Second floor decking must finish before second floor wall and corners can be framed	3_D Precedes 4_D
First floor sheathing can be installed after first floor walls and corners have been framed	2_D Precedes 6_D
Second floor sheathing can be installed after second floor walls and corners have been framed	4_D Precedes 7_D
Second floor decking can only start after first floor walls and corners have been framed	2_D Precedes 3_D
Once the second floor walls and corners have been framed, roof-framing can start	4_D Precedes 5_D
Roof-framing must be complete before roof-decking can start	5_D Precedes 8_D

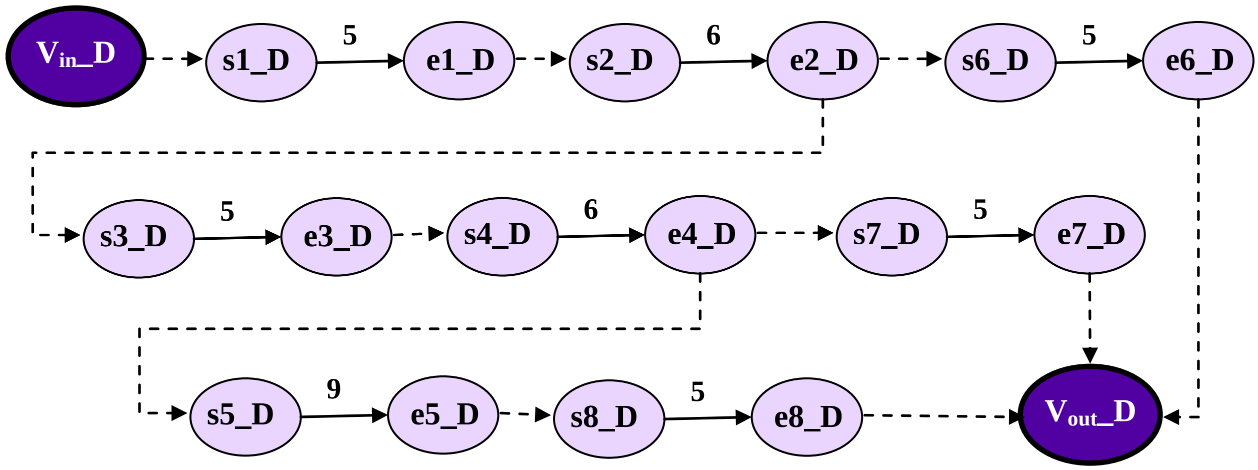


Figure 15: Detailed Point Graph for Activity D (Framing and Dry-in)

Top-Down Approach

In the high-level Point Graph for the project, the duration of activity D was estimated to be 44 units. In the top-down approach, this estimate becomes a constraint on the completion time of the sub-plan representing activity D. This constraint can be added by one of the following two ways: 1) A dummy concurrent activity with the same duration as for D (e.g., 44 days) is added to the sub-plan before the scheduling algorithm Forward* is applied to the resulting PG, and Reverse* is applied after setting the late and latest occurrence times of the virtual sink node by adding free float and total float to the earliest occurrence time, respectively; or 2) the earliest time of the virtual source node in the sub-plan is set to the earliest occurrence of the node representing the start of activity D in the high-level PG before the application of Forward*, then the late and latest times of the virtual sink node in the sub-plan are set to the corresponding attributes of the node representing the end of activity D in the high-level PG before the application of Reverse* algorithm. In case the completion time required for the sub-plan is smaller than the high-level estimate, additional floats become available to the sub-activities constituting the sub-plan. On the other hand, if the sub-plan requires longer time, then the Point Graph becomes inconsistent, indicating the infeasibility of the whole project. Table 12 shows the attributes of the sub-activities corresponding to activity D. Once a sub-plan is found feasible and the attributes of the sub-activities are calculated, top-down approach requires no further processing of the high-level Point Graph.

Table 12: Attributes for the Sub-Activities of activity D

Activity	Duration	Earliest Start Time	Latest End Time	Critical	Total Float	Free Float
V_{in}_D	0	63	63	yes	0	0
1_D	5	63	73	no	5	0
2_D	6	68	79	no	5	0
3_D	5	74	84	no	5	0
4_D	6	79	90	no	5	0
5_D	9	85	99	no	5	0
6_D	7	74	107	no	26	26
7_D	7	85	107	no	15	15
8_D	8	94	107	no	5	5
V_{out}_D	0	107	107	yes	0	0

Bottom-Up Approach

In this approach, each sub-plan with sub-activities is processed before the high-level plan. The completion times for all the sub-plans are calculated and then propagated to the high-level plan where they are used as durations of the high-level activities corresponding to the sub-plans. Once all the time calculations are done at the sub-plan level, the high-level plan is processed by the application of scheduling algorithms on the Point Graph representing the plan. The resulting time floats, if any, for the high-level activities can be propagated downwards to the activities in the corresponding sub-plans. The results of this approach for the example are illustrated in Table 13 where the duration for activity D represents the completion time of the sub-plan in Figure 15. It is apparent in the description of the two approaches that the two can be combined in a single project.

Table 13: (Recalculated) Attributes for the Activities in the Project

Activity	Duration	Earliest Start Time	Latest End Time	Critical	Total Float	Free Float
V_{in}	0	0	0	yes	0	0
A	21	0	21	yes	0	0
B	0	21	21	yes	0	0
C	42	21	63	yes	0	0
D	39*	63	102	yes	0	0
E	19	102	121	yes	0	0
F	30	102	134	no	2	2
G	13	121	134	yes	0	0

Activity	Duration	Earliest Start Time	Latest End Time	Critical	Total Float	Free Float
H	8	134	142	yes	0	0
V_{out}	0	142	142	yes	0	0

*here duration is equal to completion time of the sub-plan

Constraints on Start/End of Activities

The successful completion of a project plan also depends upon a number of additional constraints that may not be present or modeled in the initial Point Graph (or plan.) For example, late delivery of material, constraints on the availability of equipment required to undertake an activity, imposition of specific deadlines or milestones by contractors/regulating authority, and/or delays in paperwork may require an initial plan to be revised with the additional temporal constraints. These constraints may be introduced to a project plan at any stage of its inception and/or execution. In this illustration we demonstrate the handling of such additions to the project plan modeled as a Point Graph. The following are some constraints that we add to the plan in Table 7 and 8 (Figure 14.)

i) The equipment for laying foundation will be available after Day 24, i.e., Stamp [sC] $\geq$ 25;

ii) Final inspection must be completed on or before Day 155, i.e., Stamp [eH] $\leq$ 155.

The Point Graph with the added constraints is shown in Fig. 16. The application of the scheduling algorithms on the graph results in new attributes for the activities (Table 14).

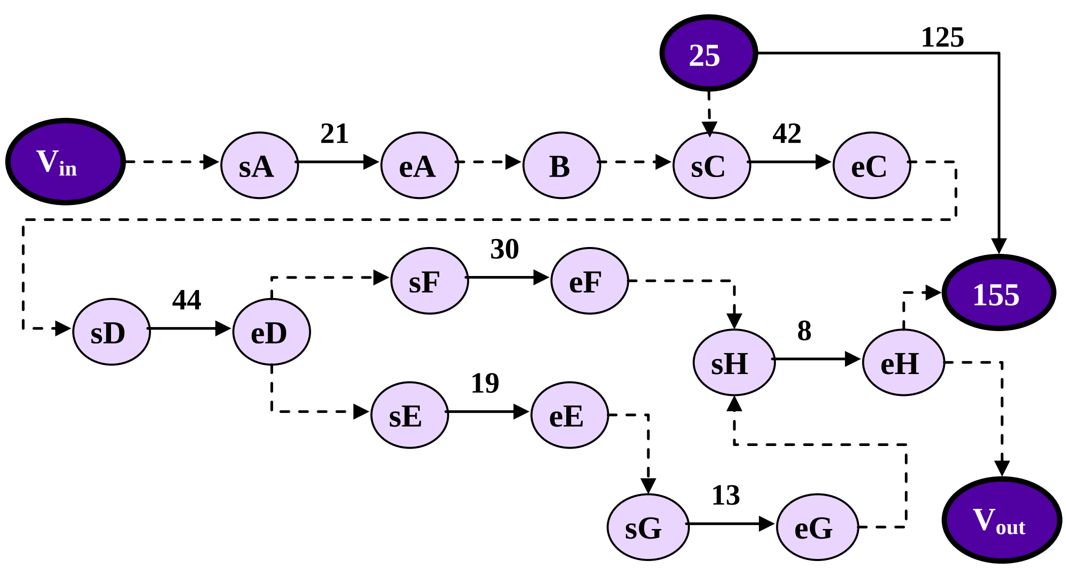


Figure 16. Point Graph with Additional Constraints

Table 14: Attributes for the Activities in the Project

Activity	Duration	Earliest Start Time	Latest End Time	Critical	Total Float	Free Float
V_{in}	0	0	0	yes	0	0
A	21	0	25	no	4	0
B	0	21	25	no	4	4
C	42	25	67	yes	0	0
D	44	67	111	yes	0	0
E	19	111	130	yes	0	0
F	30	111	143	no	2	2
G	13	130	143	yes	0	0
H	8	143	151	yes	0	0
V_{out}	0	151	151	yes	0	0

Observe in Table 14 that the scheduling algorithm tries to minimize the completion time and hence the upper bound on the end time of activity H has no effect. It is possible, however, to enforce a predetermined completion time and calculate float values accordingly.

Monitoring

Once a plan's activities start executing, the actual completion (or end) times may be earlier or later than anticipated in the original plan. The activity durations used in a plan are estimates and the actual durations may be different. The approach presented in this paper can be used to monitor a plan's execution by the use of the 'Stamp' function for the start/end of activities as soon as such information becomes available during the plan execution. Every time a specific time stamp is provided to the Point Graph, reflecting the start and/or end of some activity in the plan, the graph is processed by the scheduling algorithms to revise the time attributes associated with the remaining activities. The application of algorithms may result in new floats or new critical activities in the remaining part of the project plan. To illustrate this real-time monitoring of a project plan, we add the following two completion times to the Point Graph in Figure 16.

i) Plan finalization, estimates and signing of contract took place on Day 22, i.e., Stamp [A] = 22;
ii) Permit is issued on Day 23, i.e., Stamp [B] = 23.

The revised plan is shown in Table 15.

Table 15: Attributes for the Activities in the Revised Plan

Activity	Duration	Earliest Start Time	Latest End Time	Critical	Total Float	Free Float
V_{in}	0	0	1	no	1	1
A	21	1	22	yes	0	0
B	0	23	23	yes	0	0
C	42	25	67	yes	0	0
D	44	67	111	yes	0	0
E	19	111	130	yes	0	0
F	30	111	143	no	2	2
G	13	130	143	yes	0	0
H	8	143	151	yes	0	0
V_{out}	0	151	151	yes	0	0

6. Temper—The Software Implementation

The approach presented in the previous sections has been implemented as a software tool called Temper (Temporal Programmer). Temper provides a language editor to input PIL statements. It has a graphical interface to display the Point Graphs and also a text I/O interface to display information and results of the analyses. The algorithms are implemented in the form of a .NET class library called *PIL Engine* that provides an application programming interface (API) which can be used in any .NET compliant programming language. It uses *QuickGraph*, which is an open-source C# implementation of *Boost Graph Library* (BGL), and *Graphviz* library from AT&T (Graphviz), for internal graph representation and for implementation of PIL algorithms. In Figure 17 we show the user interface for Temper and in Table 16 we summarize the methods exposed by the application programming interface of PIL Engine.

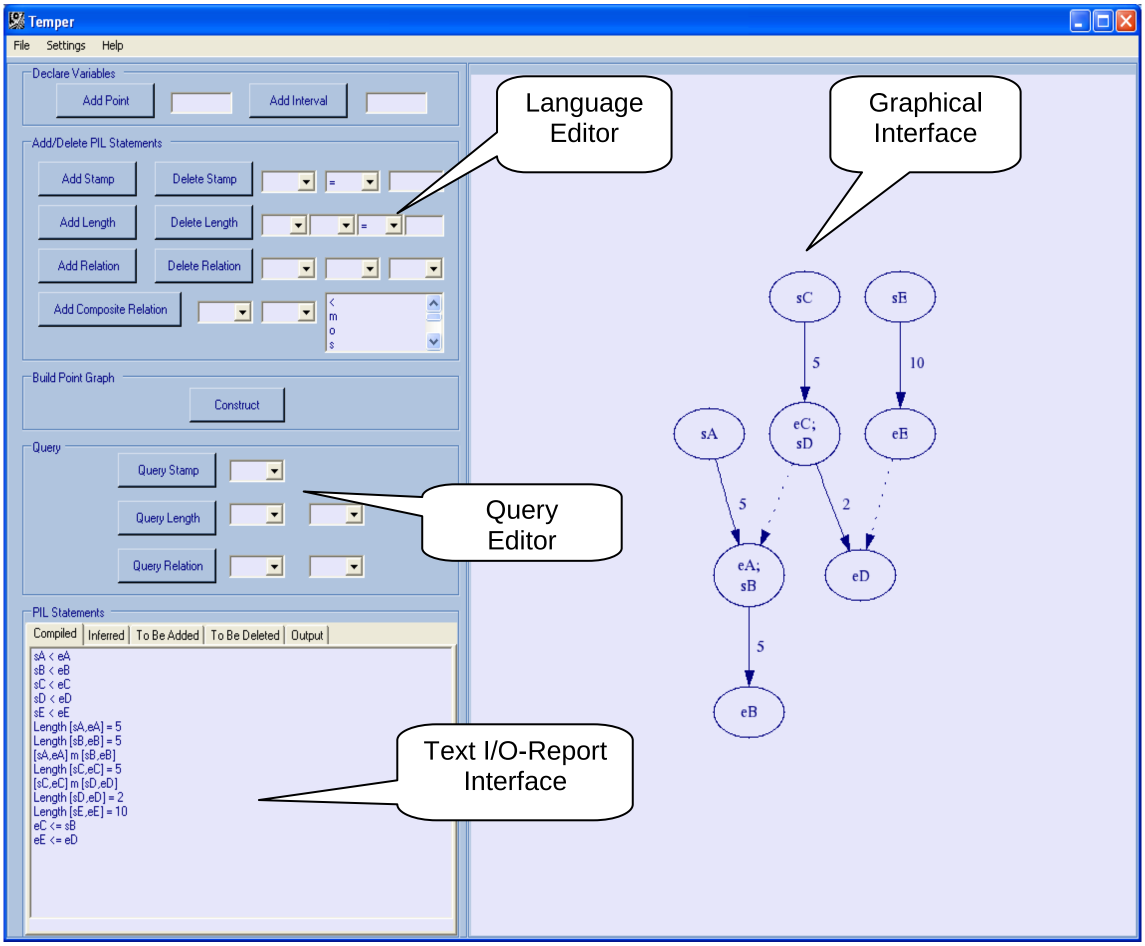


Figure 17: Temper's User Interface

Table 16: Application Programming Interface (API) of Temper

Method	Purpose
addPoint	Declares a variable of type *point*
addInterval	Declares a variable of type *interval*
addStampStatement	Assigns stamp to a point
addLengthStatement	Assigns length to an interval
addStatement	Add a relation between two temporal variables
addCompositeStatement	Add a relation containing disjunctions
deleteStampStatement	Deletes the stamp of a point
deleteLengthStatement	Deletes the length of an interval
deleteStatement	Deletes the relation between two temporal variables
queryStamp	Infers the stamp of a point
queryLength	Infer the length of an interval
queryRelation	Infers the relation between two temporal variables

Conclusion

The approach presented in this paper extends the classical duration-based quantitative approaches for project management and monitoring by adding the provision for point (instantaneous) activities and specification of partially ordered relation between system activities. It also offers an expressive input language for project managers to input their specifications. The accompanying software offers an analytic toolkit for project/system managers and engineers for planning and monitoring large scale development projects as well as for scheduling services in the modern-day system-of-systems that acquire and lose services or parts of other systems at run-time to develop new and unprecedented capabilities. In that respect, the presented approach extends the capabilities of mission planners, program managers, and project designers in the operations research and systems engineering communities for planning missions/projects beyond the traditional CPM techniques currently being used.

References

Kelley J. E., Jr. and M. R. Walker, "Critical Path Planning and Scheduling," Proc. of Eastern Joint Computer Conference, 16, 1959, 160-172.

Moder, J., and C. Philips, Project Management with CPM and PERT, 2nd Edition, Van Nostrand Reinhold, NY, 1970.

Rina Dechter, Itay Meiri, and Judea Pearl, "Temporal Constraint Networks," Artificial Intelligence, 49 (1991) 61-95.

Salah E. Elmaghraby; Jerzy Kamburowski, "The Analysis of Activity Networks under Generalized Precedence Relations (GPRs)," Management Science, Vol. 38, No. 9. (Sep., 1992), pp. 1245-1263.

Kerbosch J. A. G. M. and H. J. Schell, "Network Planning by the Extended METRA Potential Method," Report KS-1.1, University of Technology Eindhoven, Department of Industrial Engineering, Netherlands, 1975.

Salah E. Elmaghraby, "Activity Nets: A Guided Tour Through Some Recent Developments," European Journal of Operational Research 82 (1995) 383-408.

Zaidi, A. K., "On Temporal Logic Programming Using Petri Nets," IEEE Transactions on SMC, Part A, 29(3): 245-254, May 1999.

Jerzy Kamburowsky, David J. Michael, and Matthias F.M. Stallmann, "Minimizing the Complexity of an Activity Network," Networks, Vol. 36(1) 47-52 (2000). John Wiley & Sons, Inc.

Zaidi, A. K., "A Temporal Programmer for Time-Sensitive Modeling of Discrete-Event Systems," in Proc. of IEEE – SMC 2000 Meeting, Nashville, TN, October 2000.

Zaidi, A. K. and A. H. Levis, "TEMPER: A Temporal Programmer for Time-sensitive Control of Discrete-event Systems," IEEE Transaction on SMC, 31, 6, 485-496, 2001.

Erik L. Demeulemeester and Willy S. Herroelen, Project Scheduling, A Research Handbook, Kluwer, 2002.

Ulrich Dorndorf, Project Scheduling with Time Windows—From Theory to Applications, Springer, 2002.

A Guide to Project Management Body of Knowledge: PMBOK Guide - 3rd ed., Project Management Institute, Inc., PA, USA, 2004

Ishaque, S. M. M. On Temporal Planning and Reasoning with Point Interval Logic, *MS Thesis*, CS, George Mason University, VA, 2006.

Ishaque, M and A. K. Zaidi, “Time-Sensitive Planning Using Point-Interval Logic,” 10th International Command and Control Research and Technology Symposium, 2005.

Tamas Kis, "Project Scheduling: A Survey of Recent Books," Operations Research Letters, 33 (2005), 105-110, Elsevier.

Zaidi, A. K. and Lee W. Wagenhals, “Planning Temporal Events Using Point Interval Logic,” Special Issue of Mathematical and Computer Modeling, vol 43, Elsevier, 2006.

Y. Cohen, A. Sadeh, "A New Approach for Constructing and Generating AOA Networks," Journal of Engineering, Computing and Architecture, 1(1), 2007 (1-13).

Tyson R. Browning, Ranga V. Ramasesh, "A Survey of Activity Network-Based Process Models for Managing Product Development Projects," Production and Operations Management, Vol. 16, No. 2, March-April 2007, pp. 217–240.

Boost Graph Library: http://www.boost.org/

Grahpviz: http://www.graphviz.org/

MS Project: http://office.microsoft.com/en-us/templates/TC010184521033.aspx?CategoryID=CT101172331033

An Inference Mechanism for Point-Interval Logic

Mashhood Ishaque
Computer Science Department
Tufts University, Medford, MA
mishaq01@cs.tufts.edu

Faisal Mansoor and **Abbas K. Zaidi**
System Architectures Lab
George Mason University, Fairfax, VA
{fmansoor, szaidi2}@gmu.edu

Abstract

We present a new inference algorithm for Point-Interval Logic. The mechanism removes the incompleteness of previously reported inference mechanism for Point-Interval Logic. We also show how this inference mechanism can be used to prune the search space for an instance in Generalized Point-Interval Logic.

Introduction

Point-Interval Logic (PIL) is a tractable subclass of Allen's interval algebra [1] which is used for modeling temporal information. The temporal information given in the form of a set statements in PIL (conjunction of statements), is converted into graph representation called Point Graph (PG), and checked for consistency (there is a mapping on timeline that satisfies all constraints). Once we have a consistent PG, we can answer temporal queries by executing various graph search algorithms on the PG representation.

The language of PIL has been shown to capture the temporal aspects of time-sensitive mission planning [5, 9, 10], project management [3], and criminal forensics [4]. It is important to have a complete and efficient inference mechanism to effectively solve problems of interest in the mentioned application domains. In this paper we describe a new inference mechanism for PIL, which removes the incompleteness of the previous mechanism reported in [8, 10].

The paper has been organized as follows: in Section 2 we briefly describe PIL and its PG representation, in Section 3 we present the new inference mechanism; in Section 4 we show how to use the inference mechanism to prune the search space for instances in Generalized Point-Interval Logic; and finally in Section 5 we identify future research directions.

Point-Interval Logic and Point Graphs

We begin with a brief description of Point-Interval Logic and Point Graphs for making this presentation self-contained; same description can also be found in [3, 8, 9, 10]. PIL is a formal logic for reasoning with temporal events. It has two types of variables: points (events) and intervals (activities with duration). An interval X implicitly defines two points sX and eX that represent the start and end of the interval, respectively. The PIL is a pointisable logic [6], i.e. every relation between the temporal variables can be represented in terms of relationships between their start/end points. In Figure 1 we show examples of some temporal relationships between two intervals; for all possible relationships, see [3,9]. PIL also provides constructs to represent quantitative temporal information. In PIL a point variable can be assigned a stamp that represents its occurrence on the timeline. Similarly, an interval can be assigned a length that represents its duration on the timeline.

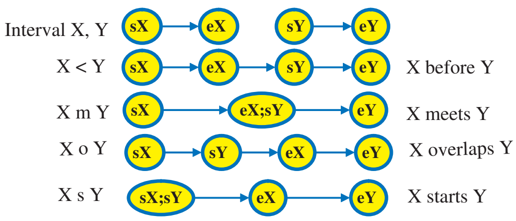


Figure 1: Interval-Interval Relationships in PIL

Point Graph is the knowledge representation scheme for Point-Interval Logic. In a PG, each node represents a point on the timeline, and edges represent the relationship between points. There can be two kinds of edges in a PG: 'less than' (LT) edge and 'less than or equal to' (LE) edge. An LT edge between two nodes p and q depicts that the point represented by the node p occurs on the timeline strictly before the point represented by q. An LE edge, on the other hand, depicts that the point corresponding to node p can either occur before or at the same moment as the point corresponding to q. In Figure 2 we show a set of PIL statements and the corresponding PG representation. In Figure 3 we show the steps involved in constructing a PG from a set of PIL statements.

Inference Mechanism

In this section, we describe the new inference mechanism for PIL. This inference mechanism is implemented in the form of various graph algorithms that operate on the PG representation of a given temporal input. These inference algorithms assume that the temporal information has already been checked for consistency. The resulting inference mechanism improves upon the previous inference mechanism for

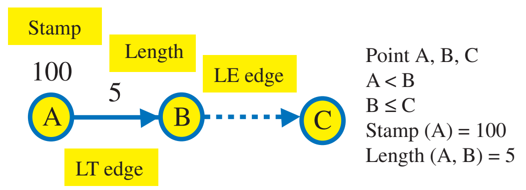


Figure 2: Point Graph for a Set of PIL Statements

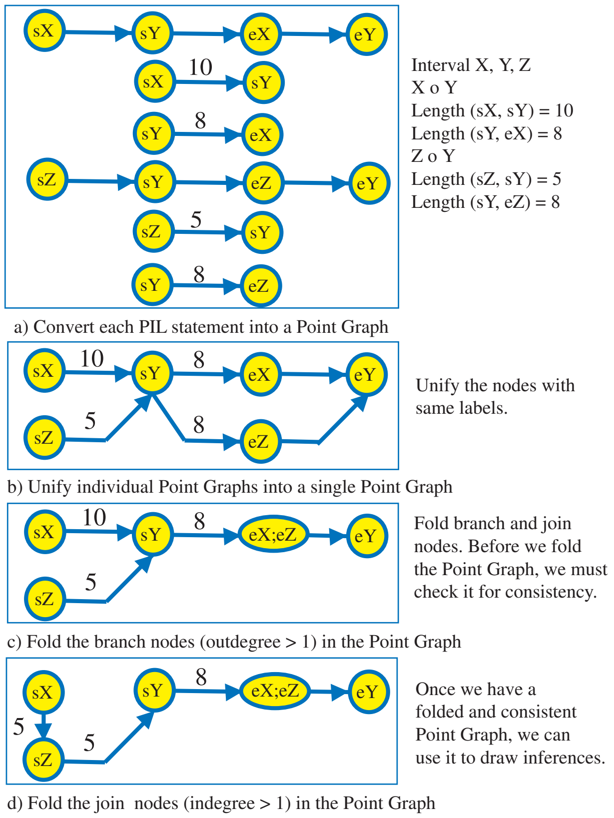


Figure 3: Steps for Constructing a Point Graph

PIL [8,10], which was not complete. The old inference mechanism was based on the idea of finding directed paths between two points in a PG, causing it to miss some temporal relations that can be inferred even in the absence of directed paths. We illustrate the incompleteness in Fig. 4.

The new inference mechanism implements the temporal queries by the following functions; all of which run in $O(m+n)$ time, where n is the number of nodes and m is the number of edges in the PG.

- **queryRelation**(p, q) - returns the relationship between the two temporal variables.
- **queryLength**(p, q) - returns the length of the interval defined by the two points.
- **queryStamp**(p) - returns the time stamp for the point.

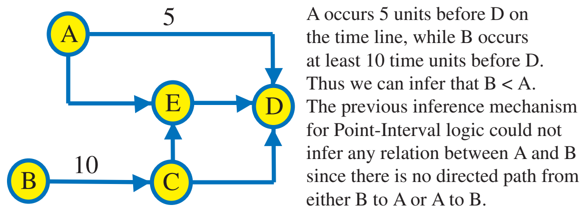


Figure 4: Illustration of Incompleteness

Inferring Relationship Between Two Variables

A relationship query asks for the temporal relationship between two variables (of type point or interval). There are three kinds of relationship queries based on type of the temporal variables.

- Inferring the relationship between two points.
- Inferring the relationship between a point and an interval.
- Inferring the relationship between two intervals.

Since PIL is a pointisable logic[6], all of the above queries can be answered using a constant number of point-point queries. For example, we can infer the relationship between two intervals $X[sX, eX]$ and $Y[sY, eY]$ by looking at the relationships among their end points. Once we have identified the relationships (sX, sY), (sX, eY), (eX, sY) and (eX, eY), we can translate this information into a set of possible relationships between the two intervals using a lookup table. Since there are six possible point-point relationships $\{\leq, <, =, \geq, >, ?\}$, the lookup table for interval-interval relationships has about $6^4 = 1296$ entries. We have computed these lookup tables manually. In Table 1 we show an snippet of the lookup table for interval-interval relationship. The subscript i denotes an inverse relationship; for example, $X f_i Y$ also means $Y f X$.

(sX, sY)	(sX, eY)	(eX, sY)	(eX, eY)	(X, Y)
$<$	$<$	$<$	$<$	$\{<\}$
$<$	$<$	$=$	$<$	$\{m\}$
$<$	$<$	$>$	$<$	$\{o\}$
$<$	$<$	$>$	$=$	$\{f_i\}$
$<$	$<$	$>$	$>$	$\{d_i\}$
$<$	$<$	$>$	$?$	$\{o, f_i\}$
$<$	$<$	$>$	$?$	$\{d_i, f_i\}$
$<$	$<$	$>$	$?$	$\{o, d_i, f_i\}$

Table 1: Lookup Table for Interval-Interval Relationships

Since we have already shown that all relationship queries can be reduced to point-point relationship queries, we will only talk about the point-point queries. The incompleteness in the previous inference mechanism resulted from the fact that the mechanism only looked for directed path between two points. But a relationship between two points might still exist even if there is no directed path. We can discover such relationships if we look for the directed path from the two query points to all nodes that are reachable from both points. We not only look for a directed path, but a directed path with the greatest length. Once we have calculated the longest

directed paths from both query points to all reachable nodes, we look at each of those nodes to see if there is sufficient information to infer a relationship. We also calculate the greatest lower bound on the path from p to q if $p < q$ or vice versa, in the process. We can find the longest path between two nodes in a directed acyclic graph using topological sort [2] in $O(m+n)$ time. Note that finding the longest path in an undirected graph is NP-Hard. For the sake of clarity and brevity we only give the version of queryRelation(p, q) that decide whether $p < q$ or not.

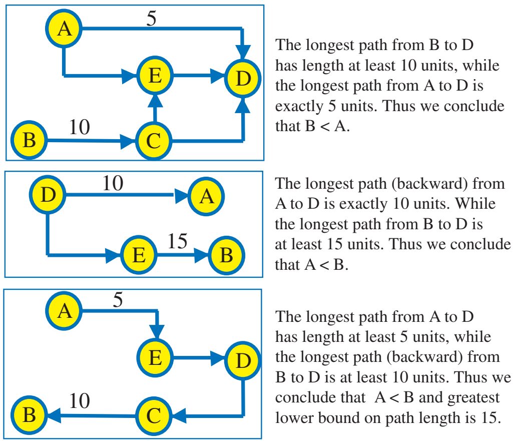


Figure 5: Inferring Relationship Between Points A and B

Since a path can contain edges with or without the length information, comparing two paths needs some explanation. We maintain for each path two parameters: the length of path so far, and a flag to indicate if the path has some edge(s) without the length information. If both the paths that are being compared, contain edges without length information, we cannot decide which path is longer. Otherwise we take the one with greater path length. If the lengths of two paths are equal and one of the paths contains a LT edge without length, then the path with LT edge is longer. While running the topological sort, the longest path is the one with the greatest length regardless of whether it contains any edge without length information. Figure 5 shows examples of how to infer the relationship between two points.

Theorem 1. *Inference mechanism correctly infers the relationship between two points, if it can be inferred at all.*

Proof. Let p and q be the two points between which we want to infer the relationship. We can have only the following four situations involving p and q:

- **p and q are not connected**
  In this case we do not have sufficient information to infer any relationship between p and q, and inference mechanism answers that relationship between p and q is unknown.
- **p has directed path to q or vice versa**
  We just consider the case when p has a directed path (with at least one LT edge without length information) to q, which implies $p < q$. Since p has directed path to q, the inference mechanism can find a node, q in this case, for which length of path from p to q is greater than that of the (zero length) path from q to q . Thus the mechanism correctly concludes that $p < q$
- **p and q have directed paths to some common node v**
  If for all such nodes v both the directed paths contain at least one LT edge without length information, we cannot conclude anything, and neither can the inference mechanism. Suppose w.l.o.g. the path from p has exact length, we can compare it with the greatest lower bound on the length of the path from q. If the length of the path from p is less than the greatest lower bound, then we can conclude that $p > q$ (farther on the timeline). Otherwise the node v does not have enough information for us to conclude anything and we look at other reachable nodes. Thus inference mechanism correctly identifies the relationship between p and q whenever sufficient temporal information is available.
- **some node v has directed paths to both p and q**
  Same argument as above.

Algorithm 1 queryRelation (p, q)

Using topological sort find the longest paths from p and q to every reachable node in the directed graph.
Again using topological sort find the longest paths from p and q to every reachable node in the directed graph obtained by reversing the direction of every edge.
Store at each node in the graph the information about the length and direction (forward/reverse) of the longest paths.
$pathLength_{glb}(p, q) \leftarrow 0$
for all nodes v in the graph **do**
  if forwardPath(p, v) and backwardPath(q, v) **then**
    $relation(p, q) \leftarrow$ '<'
    $pathLength_{lb}(p, q) \leftarrow pathLength_{glb}(p, v) + pathLength_{glb}(v, q)$
  else if forwardPath(p, v) and forwardPath(q, v) **then**
    if $pathLength_{glb}(p, v) > pathLength_{glb}(q, v)$ **then**
      $relation(p, q) \leftarrow$ '<'
      $pathLength_{lb}(p, q) \leftarrow pathLength_{glb}(p, v) - pathLength_{glb}(q, v)$
    end if
  else if backwardPath(p, v) and backwardPath(q, v) **then**
    if $pathLength_{glb}(v, p) < pathLength_{glb}(v, q)$ **then**
      $relation(p, q) \leftarrow$ '<'
      $pathLength_{lb}(p, q) \leftarrow pathLength_{glb}(v, q) - pathLength_{glb}(v, p)$
    end if
  end if
  if $pathLength_{lb}(p, q) > pathLength_{glb}(p, q)$ **then**
    $pathLength_{glb}(p, q) \leftarrow pathLength_{lb}(p, q)$
  end if
end for

The claim that the inference mechanism determines the greatest lower bound on the length of the path or exact length if it exists, comes from the ability of topological sort in finding the longest path between two nodes in a directed acyclic graph. ☐

Inferring Length of an Interval

A length query asks how far apart are the two query points on the timeline. If the exact length of the interval defined by the two points cannot be inferred, we try to find the tightest lower and upper bounds. The queryLength algorithm given here cannot be used to find the least upper bound on a path's length. For establishing the least upper bound the inference mechanism contains an exponential time algorithm, which is not discussed here. Figure 6 shows an example of how to find the length of a query interval.

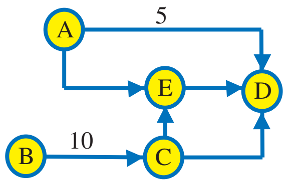


Figure 6: Inferring Length of the Interval [B, A]

Algorithm 2 queryLength (p, q)

---

queryRelation(p, q)
$Length(p, q) \leftarrow ?, Length_{glb}(p, q) \leftarrow ?$
if queryRelation returns $p < q$ **then**
  $Length_{glb}(p, q) \leftarrow pathLength_{glb}(p, q)$
  if $pathLength_{glb}(p, q)$ is exact **then**
    $Length(p, q) \leftarrow Length_{glb}(p, q)$
  end if
end if

Theorem 2. *Inference mechanism finds the exact length of interval if it can be inferred. Also the algorithm establishes the tightest lower bound for the interval length.*

Proof. The proof follows from the correctness of queryRelation algorithm. ☐

Inferring Stamp of a Point

A stamp query asks where the query point (event) occurs on the timeline. If the exact stamp of the given point cannot be inferred, we answer the query is in the form of tightest lower and upper bounds. Figure 7 shows an example of how to find the stamp of a query point.

Theorem 3. *Inference mechanism is complete for stamp queries i.e. if the time stamp of a point can be inferred, the queryStamp algorithm find that time stamp. Also the algorithm establishes the tightest lower and upper bounds for time stamp of a point.*

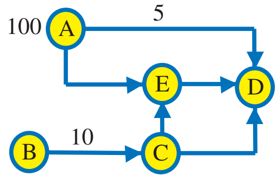


Figure 7: Inferring Stamp of the Point B

Algorithm 3 queryStamp (p)

Find the nodes s and t with smallest and largest time stamp respectively using breadth first search.
$Stamp(p) \leftarrow ?, Stamp_{glb}(p) \leftarrow ?, Stamp_{lub}(p) \leftarrow ?$
queryRelation (p, s)
if queryRelation returns $p > s$ **then**
  $Stamp_{glb}(p) \leftarrow Stamp(s) + pathLength_{glb}(s, p)$
  if $pathLength_{glb}(s, p)$ is exact **then**
    $Stamp(p) \leftarrow Stamp_{glb}(p)$
    return
  end if
else
  $Stamp_{glb}(p) \leftarrow Stamp(s) + pathLength_{glb}(s, p)$
  return
end if
queryRelation (p, t)
if queryRelation returns $p < t$ **then**
  $Stamp_{lub}(p) \leftarrow Stamp(t) - pathLength_{glb}(p, t)$
end if

Proof. A node corresponding to a query can have an exact time stamp iff there is a directed path of exact length between itself and some node with a time stamp or it has a stamp itself. Since in a consistent and folded PG all the nodes with stamps form a directed path with exact length, a query node can have an exact stamp iff there is a directed path of exact length between itself and the node with the smallest time stamp. Since we have already established the correctness of queryRelation in finding an exact length path between two nodes whenever it exists, we can conclude that queryStamp always find the exact stamp of a node whenever possible.

We now prove that the algorithm finds the tightest lower and upper bounds. Let s and t be the nodes with the smallest and largest time stamps respectively. If the query point occurs before s, we cannot infer any lower bound on the time stamp (unless we can infer an exact stamp in which we are not interested in lower and upper bounds). Similarly no upper bound can be inferred for a point that occurs after t. Thus we find the lower bound on the time stamp of the point that occurs after s by adding the greatest lower bound on the length of the path from s to query point and the time stamp of s. The correctness follows from the fact that the queryRelation always correctly discovers the greatest lower bound on the length of a path. Similar argument applies to the calculation of least upper bound on the time stamp. ☐

Generalized Point-Interval Logic

PIL is a tractable subclass of Allen's Interval Algebra. But the tractability comes at the cost of expressiveness. For example, the disjointedness relation between two intervals cannot be expressed in PIL. To overcome this deficiency we can generalize PIL by allowing statements in logic to contain disjunctions between relations; we refer to this logic as Generalized Point-Interval Logic (GPIL). Table 2 shows an instance of GPIL. Deciding consistency in GPIL is NP-Complete, since GPIL subsumes Allen's Interval Algebra [1] (hence NP-Hard), and solution to a GPIL instance can be verified in polynomial time (hence NP).

Interval A, B, C
(A m B) or (A o B) or (A d B)
(C < B) or (C > B) (disjointedness constraint)
(C s A) or (C f A)
(A < D) or (A > D)

Table 2: An Instance of Generalized Point-Interval Logic

CMI Algorithm

Assuming P≠NP we cannot expect to find a polynomial time algorithm to decide whether the given instance of GPIL is consistent. Instead we suggest a modified branch-and-bound algorithm called "Consistency Maintaining Inference-Based (CMI) Algorithm" to explore the search space (takes exponential time in the worst case). The CMI algorithm uses the inference mechanism of PIL, as the bounding function to prune the search space. Given a set of statements in GPIL, the CMI algorithm incrementally adds one PIL statement for each GPIL statement while always maintaining a consistent (hence the name) set of PIL statements.

Suppose we are given a set $S = \{s_1, s_2, ..., s_n\}$ of GPIL statements, and the CMI algorithm is about to process the GPIL statement s_{i+1}. We have a consistent set of PIL statements C_i corresponding to statements from s_1 to s_i. Suppose the statement s_{i+1} is of the form $(X r_1 Y)$ or $(X r_2 Y)$ or ... $(X r_k Y)$ where X and Y are temporal variables and $\{r_1, r_2, ..., r_k\}$ are relations. The CMI algorithm queries the inference mechanism on C_i for the set of possible relationships between X and Y. The intersection of the set of possible relations from inference engine with the set $\{r_1, r_2, ..., r_k\}$ gives the set of relations that can be added to C_i while still maintaining consistency. We can pick any relation say r_j from this intersection set and add the PIL statement $X r_j Y$ to C_i to get C_{i+1}. If we can proceed in this manner all the way to C_n it implies the GPIL instance is consistent and C_n is *a* satisfying assignment.

However, if the intersection set is empty we cannot proceed and we need to backtrack to statement s_i. We delete the last PIL statement added to C_i, and then add the PIL statement corresponding to next relation from the intersection set of the previous stage. If the intersection set at the previous have been completely explored we backtrack even further. If we cannot backtrack any further then the given GPIL instance is inconsistent. It should be pointed out that revising (adding/deleting) a set of PIL statements is very efficient [7].

Algorithm 4 CMI (S, i)

{The algorithm is invoked by calling CMI $(S, 1)$, and returns true if the instance is satisfiable}
$(X, Y) \leftarrow$ Variables in s_i
$R_1 \leftarrow$ Relations in s_i
$R_2 \leftarrow$ queryRelation (X, Y)
sort $(R1 \cap R2)$
{sort according to desired heuristics}
for all relation $r_j \epsilon (R1 \cap R2)$ **do**
  addStatement(X, r_j, Y)
  if $i = n$ **then**
    return true
  else if CMI $(S, i + 1)$ **then**
    return true
  else
    deleteStatement (X, r_j, Y)
  end if
end for
return false

Figure 8 shows the search space for instance in Table 2. The green nodes are those for which the set of relations corresponding to the path from the root form a consistent set. A path from root to a green leaf node represent a satisfying assignment. The yellow (double-circles) nodes are the first nodes on any path from root that makes the path inconsistent. CMI algorithm is forced to backtrack upon reaching a yellow node. Notice the red (smaller-circles) nodes are never explored, and thus the inference mechanism prunes the search space.

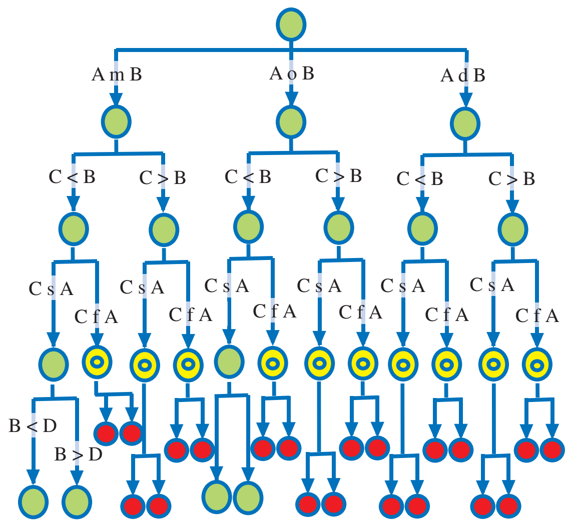


Figure 8: Search Space Exploration

Incorporating Heuristics in CMI Algorithm

CMI algorithm only explores the pruned search space (green and yellow nodes). The order in which these reachable nodes are explored can be controlled by plugging-in any

desired heuristics in CMI algorithm. A heuristics can either determine the order in which the statements of S are processed (inter-statement), or the order in which relations within an statement are added (intra-statement). To incorporate an intra-statement heuristics with CMI algorithm, sort the statements in S accordingly and then run the CMI algorithm. To plug-in an intra-statement heuristics in CMI algorithm, change the function used for sorting $(R_1 \cap R_2)$. Here we describe two intra-statement heuristics that we used for the experimental evaluation of CMI algorithm.

Succeed-First heuristics selects the least-constrained relation first. A relation is less constrained when it contains less information. For example the relation '$<$' is less constrained than the relation 'm'. The ordering of the relations from least-constrained to most-constrained is given as:

$$\{\leq, <, o, d, m, f, s, =\}$$

Fail-First heuristics selects the most-constrained relation first. The ordering of the relations is just the reverse of succeed-first ordering.

Experimental Evaluation of CMI Algorithm

We evaluated the performance of CMI algorithm on a set of randomly generated but satisfiable instances of GPIL. We compared the number of nodes in the search space explored by the CMI algorithm (with the two heuristics) until it found the first feasible solution, to the number of nodes in the search space. We present the summary of our results in Table 3, where each row is the average of about 20 random instances.

Var./Stat.	**Stat.**	**Succeed-First**	**Fail-First**	**Search Space**
50	30	1382	5894	$3.8 * 10^7$
50	40	144741	122022	$3.8 * 10^9$
75	30	66864	49277	$1.9 * 10^8$
75	40	254002	1374951	$5.1 * 10^{10}$
100	30	14382	16771	$5.1 * 10^8$
100	40	40	3226	$8.3 * 10^{10}$
125	30	30	30	$7.8 * 10^7$
125	40	40	40	$7.3 * 10^{10}$

Table 3: Number of Nodes Explored by CMI Algorithm

First column in the table represents the number of variables as a percentage of number of statements. Second column show the number of statements. The third and the fourth column represent the number of nodes explored by the Succeed-First and Fail-First heuristics, respectively. Notice that Succeed-First usually performs better that Fail-First. Also notice as we increase the ratio of number of variables to number of statements in a random instance, it becomes easier to find a feasible solution, which is intuitive since a less-constrained instance has more feasible solutions. The last column represent the total number of nodes in the search space.

Conclusions and Future Directions

In this paper we described a new inference algorithm for Point-Interval Logic. Since language of PIL has already been shown to have applications in mission planning, project management, and criminal forensics, it is important to have a complete and efficient inference mechanism for it. The presented inference mechanism is complete, though the algorithm to establish the least upper bound on the length of an interval has exponential time complexity. So one obvious open problem is to find a polynomial time algorithm or show that finding the least upper bound is NP-Hard. The other direction of work would be to introduce new queries that are built on the top of the basic relationship, length and stamp queries, and to explore how various data structural techniques can be used to speed up the query algorithms at the cost of additional preprocessing. Also it would be interesting to explore the performance of CMI algorithm on instances resulting from real-life applications.

Acknowledgments

This work was supported by the Air Force Office of Scientific Research (AFOSR) under Grants FA9550-05-1-0106 and FA9550-05-1-0388.

References

[1] Allen, J. F. 1983. Maintaining Knowledge About Temporal Intervals. *Communications of ACM* 26, 832-843.

[2] Cormen, T. H., and C. E. Leiserson, and R. L. Rivest, and C. Stein, 2nd ed. 2001. Introduction to Algorithms. MIT Press and McGraw Hill.

[3] Ishaque, M; A. K. Zaidi, and A. H. Levis 2007. Project Management Using Point Graphs. In Proceedings of 5th Conference on Systems Engineering Research.

[4] Ishaque, M, and A. K. Zaidi, and A. H. Levis 2006. On Applying Point-Interval Logic to Criminal Forensics. In Proceedings of Command and Control Research and Technology Symposium, 2006.

[5] Ishaque, M and A. K. Zaidi 2005. Time-Sensitive Planning Using Point-Interval Logic. In Proceedings of 10th International Command and Control Research and Technology Symposium.

[6] Ladkin, P. B., and Maddux, R. 1988. On binary constraint networks, Technical Report, KES.U.88.8, Kestrel Institute, Palo Alto, Calif.

[7] Rauf, I., and Zaidi, A. K. 2002.On Revising Temporal Models of Discrete-Event Systems. In Proceedings of 2002 IEEE International Conference on Systems, Man, and Cybernetics, Hemmamat, Tunisia.

[8] Zaidi, A. K. and A. H. Levis 2001. TEMPER: A Temporal Programmer for Time-sensitive Control of Discrete-event Systems. *IEEE Transaction on SMC*: 31, 6, 485-496.

[9] Zaidi, A. K. and Lee W. Wagenhals 2006. Planning Temporal Events Using Point Interval Logic. *Special Issue of Mathematical and Computer Modeling*: 43, Elsevier, 1229-1253.

[10] Zaidi, A. K. 2001. A Temporal Programmer for Time-Sensitive Modeling of Discrete-Event Systems. In Proceedings of IEEE SMC 2000 Meeting, Nashville, TN, 2186-2191.

GMU/SAL 06-201

Using Temporal Reasoning for Criminal Forensics against Terrorists*

Abbas K. Zaidi

Mashhood Ishaque

Alexander H. Levis

Point of Contact: **Abbas K. Zaidi**
Address: System Architectures Laboratory
ECE Dept., MSN 1G5
George Mason University
Fairfax, VA 22030
703.993.1725 (v)
703.993.1601 (f)

Email: szaidi2@gmu.edu

System Architectures Laboratory

USING TEMPORAL REASONING FOR CRIMINAL FORENSICS AGAINST TERRORISTS[1]

Abbas K. Zaidi

Mashhood Ishaque

Alexander H. Levis

System Architectures Lab
George Mason University
Fairfax, VA – 22030
<szaidi2, alevis>@gmu.edu

Abstract

The paper presents an application of temporal knowledge representation and reasoning techniques to forensic analysis, especially in answering certain investigative questions relating to time-sensitive information about a terrorist activity. A brief introduction to a temporal formalism called Point-Interval Logic is presented. A set of temporal facts is taken from the London bombing incident that took place on July 7, 2005, to illustrate the use of temporal reasoning for criminal forensics. The information used in the illustration is gathered through the online news sites. A hypothetical investigation on the information is carried out to identify certain time intervals of potential interest to counter-terrorist investigators. A software tool called Temper that implements Point-Interval Logic is used to run the analyses and reasoning presented in the paper.

[1] The work was carried out with support provided by the Air Force Office of Scientific Research under contract numbers FA9550-05-1-0106.

1. INTRODUCTION

While a sequence of events may unfold linearly in time, information about them comes in segments from different locations at different times, often overlapping and often with small contradictions,. This phenomenon occurs at a centralized information gathering node where information can be analyzed and fused both because of the different times sensors make their information available and because of the paths that the data or information takes to reach the centralized node. As the US Department of Defense and the Intelligence Community move towards an information sharing paradigm in which data and information are published by all entities and subscribers can access them, the need to develop algorithms that acknowledge this paradigm and even exploit it gain in importance. In colloquial terms, the question is how quickly can we connect the dots when different dots arrive in random order and at random times. We need to know when we have enough dots (given the information that they carry) to declare that we can make a useful inference – that they have been connected.

This problem has particular significance when one tries to reconstruct the sequence of events that led to an observable effect and, especially, to identify the time interval during which some critical activity has taken place. This can be thought of as forensic analysis of a set of given data. The problem becomes more challenging, if the process is undertaken while pieces of information are arriving and there is a time sensitive aspect to it, i.e., useful inferences need to be made as quickly as possible.

Consider, for example, information regarding events surrounding some criminal activity or an act of terrorism to be unfolding in no specific order. The information gathered at some instant in time, in turn, may be incomplete, partially specified, and possibly inaccurate, or inconsistent, making it difficult for investigators and counter-terrorism experts to piece together the events that can help resolve some of the investigative questions. The time-sensitive information, the information about the timing of events surrounding a criminal/terrorist act, may contain hidden patterns or temporal relations that can help identify missing links in an investigation. This calls for a formal, computer-aided approach to such an analysis.

The growing need for a formal logic of time for modeling and analyzing temporal information has led to the emergence of various types of representations and reasoning schemes, extensively reported in the research literature. This paper demonstrates the use of one such formalism, called Point-Interval Logic (PIL), in addressing the problems listed above; or how it can be used to create temporal models of situations arising in forensics and help investigators operating in real time answer interesting questions in a timely manner. An earlier student paper by Ishaque et al. (2006) first proposed and demonstrated the application of PIL to criminal forensics.

This paper is organized as follows: Section 2 gives a very brief overview of some of the influential work on temporal reasoning and knowledge representation. Section 3 presents an informal description of Point-Interval Logic (PIL). This section gives a description of the logic's syntactic and semantic structure, presents a graphical representation for the temporal statements, and illustrates inference mechanism and algorithm for deciding consistency. The software implementation of the approach, called Temper, is also discussed in this section. A hypothetical investigation on the information taken from London Bombing of July 07, 2005, is carried out with the help of Temper to identify certain time intervals of potential interest to counter-terrorism experts. It also shows that the approach not only operates on the arriving sequence of

data but also, as a result of the diagnostics produced by the inference engine, it identifies in a visual manner the type of information that is needed to disambiguate the inferences.

2. TEMPORAL KNOWLEDGE REPRESENTATION AND REASONING

The earliest attempts at formalizing a time calculus date back to 1941 by Findlay, and 1955 by Prior. Since then, there have been a number of attempts on issues related to this subject matter, like topology of time, first-order and modal approaches to time, treatments of time for simulating action and language, etc. The development of some of these formalisms has matured enough to attract comparative analyses for the computational aspects of these calculi and their subclasses. A number of researchers have attempted to use temporal reasoning formalisms for planning, plan merging, conditional planning, and planning with uncertainty problems. Other applications of temporal logics include specification and verification of real-time reactive planners, and specification of temporally-extended goals and search control rules.

As noted by Kautz (1999), the work in temporal reasoning can be classified in three general categories: algebraic systems, temporal logics, and logics of action. The work on the temporal reasoning within the framework of constraint satisfaction problems was initiated with the influential Interval Algebra (IA) of Allen (1983) (formalized as an algebra by Ladkin and Maddux, 1994) and was followed by Vilain and Kautz's (1986) Point Algebra (PA). They also proved that the problem of determining consistency in IA is NP-Complete. The nature of temporal variables, i.e., point and/or interval, and different classes of constraints, namely qualitative and/or quantitative constraints, have led to a characterization of different types of temporal formalisms: qualitative point, quantitative point, qualitative interval, quantitative interval, and their combinations. A temporal relation (or constraint) between two variables X_i and X_j is represented by the expression 'X_i C_{ij} X_j', where $C_{ij} = \{r_1, r_2, \ldots, r_n\}$; r_i's are basic relations and the expression translates to $(X_i\ r_1\ X_j) \vee (X_i\ r_2\ X_j) \vee \ldots \vee (X_i\ r_n\ X_j)$. A relation $r_i \in C_{ij}$ is *feasible* for the time variables, if and only if there exists one *solution* holding the relation between the two variables. A *solution* to such a temporal problem consists of finding *consistent* relations among all the variables involved. The notion of consistency is defined with respect to specific semantics for each type of temporal problem and with the help of *transitivity/composition tables*. A computationally less expensive notion of *local-consistency* is employed in most implementations to incrementally achieve global consistency. In most cases, enforcing local consistency can be done in polynomial time. Almost all the temporal formalisms use some form of graph representation, e.g., constraint networks, distance graphs, timegraphs, point graphs, etc., for representing/formulating the temporal problem under consideration. Some of these graphs are merely used for representing the variables and temporal relations between them, whereas others exploit the graph-theoretic properties to enforce consistency and process queries.

Point-Interval Logic (PIL) is a specialization of Pointisable Algebra (Ladkin and Maddux 1988), which is the first and the simplest tractable subclass identified, containing all the basic temporal relations, [Vilain et al., 1990]. This class is characterized by the fact that the temporal relations in it can be represented by specifying relations between the *start/end* points of intervals. A time interval X is defined with the help of a pair of its starting point X^- and end point X^+. All allowable temporal relations can be represented as constraints between these start and end points associated with time intervals. Polynomial time algorithms for processing Pointisable algebra have been developed [Gerevini and Schubert, 1993; Drakengren and Jonsson, 1997]. The work

on PIL originated from an earlier work on temporal knowledge representation and reasoning by Zaidi (1999). A graph model, called Point Graph (PG), is shown to represent the temporal statements in this approach. An inference engine based on this Point Graph representation infers new temporal relations among system intervals, identifies temporal ambiguities and errors (if present) in the system's specifications, and finally identifies the intervals of interest defined by the user. Zaidi and Levis (2001) further extended the point-interval approach by adding provisions for *dates/clock* times and time *distances* for points and intervals. This extension allowed the assignment of actual lengths to intervals, time distances between points, and time stamps to points representing the actual time of occurrences, whenever such information is available. A temporal model may change during and/or after the system specification phase. Support for an on-the-fly revision (add, delete, modify) was added to Point Graph formalism in Rauf and Zaidi (2002). Zaidi and Wagenhals (2006) consolidated the results of the previous work on the logic and its application to the modeling and planning time-sensitive aspects of a mission and extended the approach further. The extension allows for a larger class of temporal systems to be handled by incorporating an enhanced input lexicon, allowing increased flexibility in temporal specifications, providing an improved verification and inference mechanism, and adding a suite of analysis tools.

3. POINT INTERVAL LOGIC

This section presents a brief introduction to the PIL formalism with the help of illustrative examples. A more technical and detailed description can be found in Zaidi & Wagnehals (2006), Ishaque (2006), and/or Zaidi and Levis (2001).

3.1 Language and Point Graphs Representation

The lexicon of the Point Interval Logic (PIL) consists of the following primitive symbols:

Points: A point X is represented as [pX, pX] or simply [pX].

Intervals: An interval X is represented as [sX, eX], where 'sX' and 'eX' are the two end points of the interval, denoting the 'start' and 'end' of the interval, such that sX < eX.

Point Relations: These are the relations that can exist between two points. The set of relations R_P is given as:

$$R_P = \{<, =, \leq\} \text{ or } R_P = \{\textit{before, equals, precedes}\}$$

Interval Relations: These are the atomic relations that can exist between two intervals. The set of relations R_I is given as:

$$R_I = \{<, m, o, s, d, f, =\} \text{ or}$$

$$R_I = \{\textit{before, meets, overlaps, starts, during, finishes, equals}\}$$

Point-Interval Relations: These are the atomic relations that can exist between a point and an interval. The set of relations R_{PI} is given as:

$$R_{PI} = \{<, s, d, f\} \text{ or } R_{PI} = \{\textit{before, starts, during, finishes}\}$$

The symbol '?' is used to represent an unknown relationship.

Functions: The following two functions are used to represent quantitative information associated with intervals:

The *Interval length function* assigns a non-zero positive real number to a system interval.

Length X = d, where X = [sX, eX], $d \in \Re^{+}$

This function is also used to assign lower and upper bounds to an interval length. The two bounds can also be seen as representing *at least* and *at most* temporal relations.

Length $X \geq d$, where X = [sX, eX], $d \in \Re^{+}$ (d is a lower bound on length)

Length $X \leq d$, where X = [sX, eX], $d \in \Re^{+}$ (d is an upper bound on length)

The *stamp function* similarly assigns a non-negative real number to a point, or lower and upper bounds to it. The two bouns can also be seen as representing *no later than*, and *no earlier than* temporal relations.

Stamp p = t, where $t \in \Re$ $(= \Re^{+} \cup \{0\})$

Stamp $p \leq t$, $t \in \Re$

Stamp $p \geq t$, $t \in \Re$

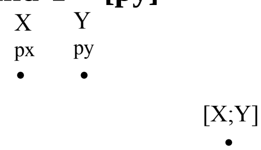
Table 3.1 shows the syntactic and semantic structure of PIL expressions. Note that each relationship between intervals or an interval and a point can be constructed with the help of inequalities between their start and end points, and by assigning values to expressions involving these points.

Table 3.1 PIL Expressions and Their Semantics

Qualitative Relations

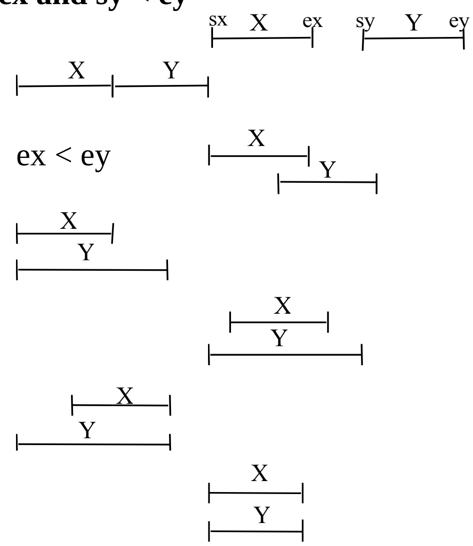
CASE I—X and Y both points: X = [px] and Y = [py]

1.	X < Y	px < py
2.	X = Y	px = py
3.	X ≤ Y	px ≤ py



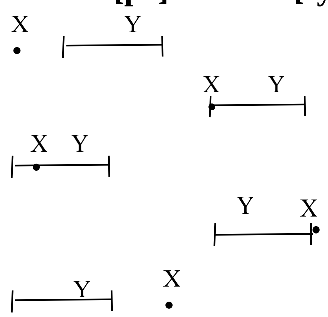
CASE II— X and Y both intervals with non-zero lengths:
X = [sx, ex], Y = [sy, ey] with sx < ex and sy < ey

1.	X < Y	ex < sy		
2.	X m Y	ex = sy		
3.	X o Y	sx < sy,	sy < ex,	ex < ey
4.	X s Y	sx = sy,	ex < ey	
5.	X d Y	sx > sy,	ex < ey	
6.	X f Y	sx > sy,	ey = ex	
7.	X = Y	sx = sy,	ex = ey	



CASE III— X is a point and Y is an interval: X = [px] and Y = [sy, ey]

1.	X < Y	px < sy
2.	X s Y	px = sy
3.	X d Y	sy < px < ey
4.	X f Y	px = ey
5.	Y < X	ey < px

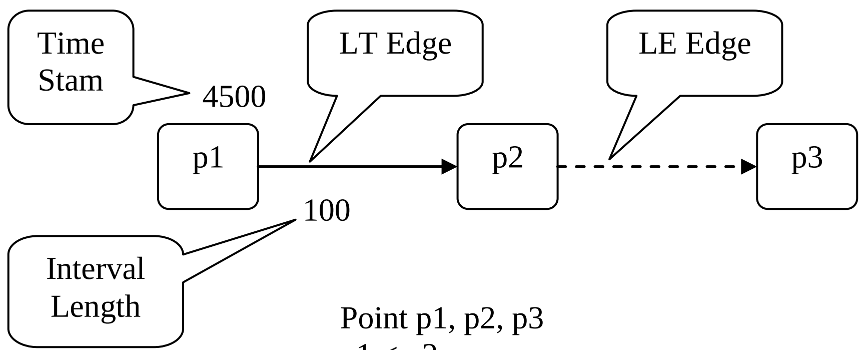


Quantitative Relations

X is a point and Y is an interval: X = [px] and Y = [sy, ey]

1. Length Y = d ey – sy = d   2. Length Y ≥ d ey – sy ≥ d
3. Length Y ≤ d ey – sy ≤ d
4. Stamp X = d px = d   5. Stamp X ≥ d px ≥ d
6. Stamp X ≤ d px ≤ d

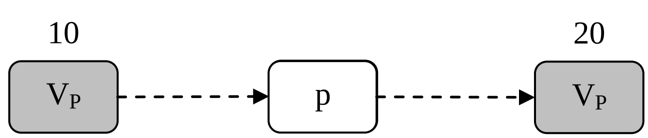
A graph construct called Point Graphs (PG) is used as an underlying structure to represent statements in PIL. In a PG, a node represents a point (or a *composite* point) and an edge between two points represents one of the two temporal relations, *before* and *precedes*, between the two. Two or more points pi, pj, …, pn are represented as a composite point [pi; pj; …; pn], or a single node in a PG, if all are mapped to a single point on the timeline. The statements in PIL can be converted to an equivalent PG representation with the help of the corresponding analytic inequalities shown in Table 3.1. In addition, the quantitative temporal information, modeled using the length and the stamp functions, is represented as node and arc inscriptions on the PG. All the verification, revision, and inference algorithms work by manipulating this Point Graph representation of the set of PIL statements. Figure 3.1 shows a set of PIL statements and the corresponding Point Graph representation.



Point p1, p2, p3
p1 < p2
p2 ≤ p3
Stamp p1 = 4500
Length [p1, p2] = 100

Figure 3.1: Point Graph Representation of a Set of PIL Statements

The Point Graph (PG) in Figure 3.1 illustrates how the inequalities shown in Table 3.1 for each qualitative temporal relation can be translated to a corresponding equivalent graph structure. The figure also shows graphical objects representing some of the quantitative temporal relations. The graphical representation of the remaining temporal relations requires introduction of virtual time point(s) or *virtual nodes* in a PG. A virtual node is like any other node in a PG except for the fact that there is no temporal variable (point, start of interval, or end of interval) associated with it. It, therefore, does not have a unique identifier or *name* associated with it. Figures 3.2 and 3.3 illustrate the PG representations of the quantitative temporal cases not covered by the structure in Figure 3.1.



Point p
Stamp p ≥ 10
Stamp p ≤ 20

Figure 3.2: PG Representation of Lower and Upper Bounds on Stamps

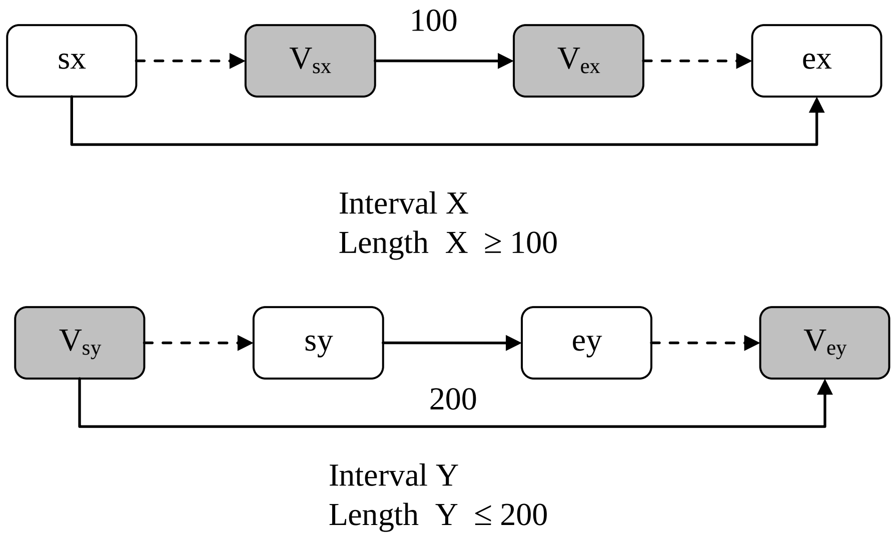


Figure 3.3: PG Representation of Lower and Upper Bounds on Interval Lengths

A formal definition of Point Graphs is given as follows:

<u>Definition:</u> Point Graphs

A Point Graph, PG (V, E_A, D, T) is a directed graph with:

V: Set of vertices with each node or vertex $v \in V$ representing a point on the real number line. Points pi, pj, …, pn are represented as a composite point [pi; pj; …; pn] if all are mapped to a single point on the line.

E_A: Union of two sets of edges: $E_A = E \cup E_{\leq}$, where

- E : Set of edges with each edge $e_{12} \in E$, between two vertices v1 and v2, also denoted as (v1, v2), representing a relation '<' (*before*) between the two vertices—(v1 < v2). The edges in this set are called LT edges;
- $E_{\leq}$: Set of edges with each edge $e_{12} \in E_{\leq}$, between two vertices v1 and v2, also denoted as (v1, v2), representing a relation '≤' (*precedes*) between the two vertices—(v1 ≤ v2). The edges in this set are called LE edges.

D: Edge-length function (possibly partial): $E \rightarrow \Re^+$

T: Vertex-stamp function (possibly partial): $V \rightarrow \Re$

3.2 Operations On Point Graphs

A relation Ri between two intervals X and Y can now be represented by an equivalent Point Graph representation by translating the algebraic inequalities/expressions shown in Tables 3.1 to corresponding PGs, as illustrated in Figures 3.1-3. The PG representing a set of PIL statement is then constructed by *unifying* individual PGs to a (possibly) single connected graph. The unifying process looks at the labels of the nodes (except for virtual nodes) and the values of the stamps associated with them to identify equalities. The nodes identified as being equal to one another are merged into a single node with a composite label. The *unified* PG is then *folded* with the help of lengths on edges. This folding process establishes new relations among system intervals, inferred through the quantitative analysis of the known relations specified by interval lengths and stamps. Figure 3.4 illustrates the two operations, unification and folding, on an example set of PIL

statements. A more technical and detailed description of the two processes can be found in Zaidi and Wagnehals (2006), Ishaque (2006), or Zaidi and Levis (2001).

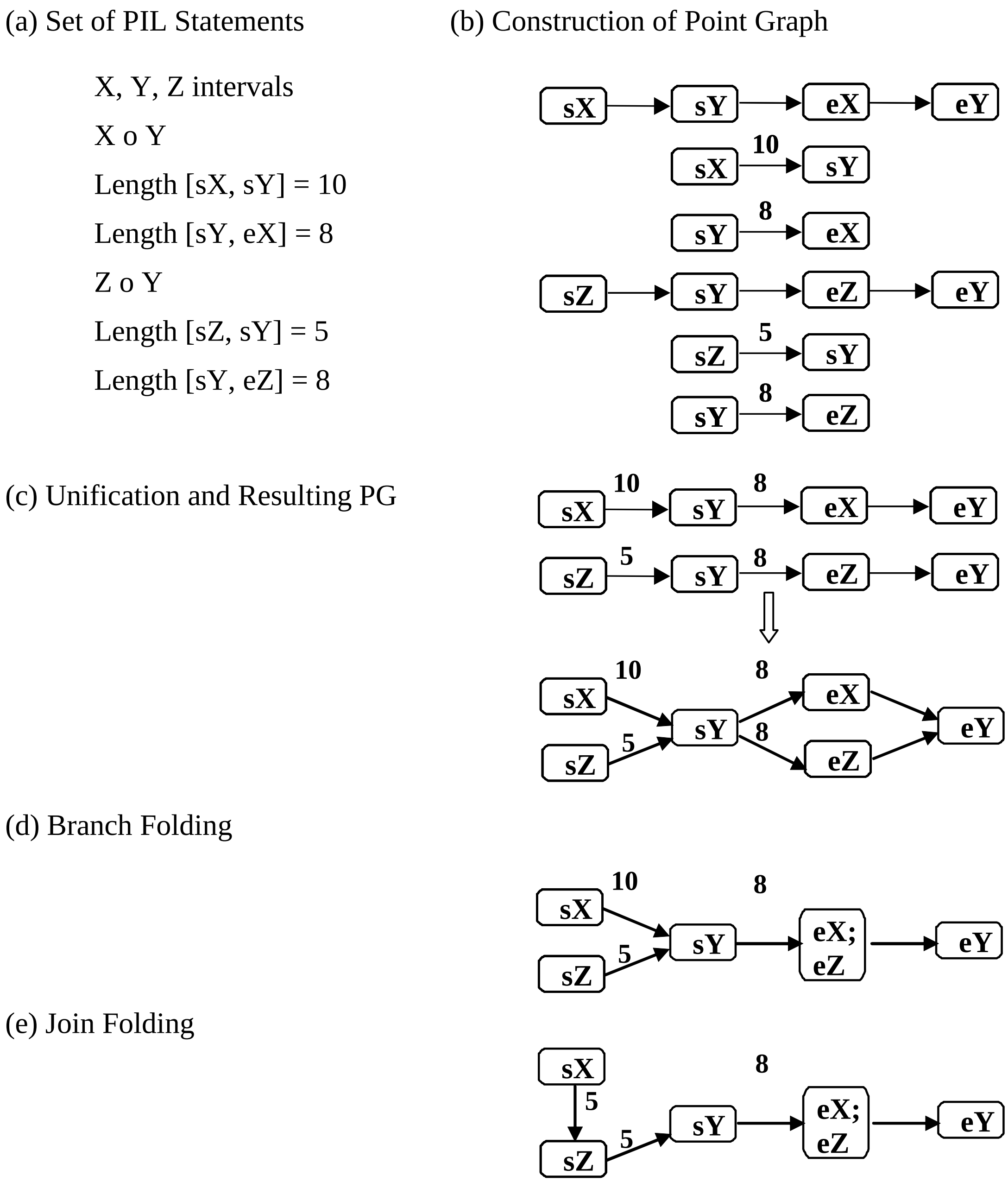


Figure 3.4: Steps in PG Construction

3.3 Inference

Two points, p1 and p2, on a real number line are related to each other by one of the following three algebraic relations: '<' (*before*), '=' (*equals*), and '≤' (*precedes*). A relation Ri between two intervals X and Y, denoted as 'X Ri Y' can, therefore, be represented as a 4-symbol string 'abcd' made of elements from the alphabet {<, =, >, ≤, ≥, ?}, where the first (left-most) symbol

‘a’ represents the algebraic relation between sX and sY, second symbol ‘b’ between sX and eY, third symbol ‘c’ represents relation between eX and sY, and fourth ‘d’ between eX and eY. The ‘?’ is added to incorporate incomplete information. The inequalities in Table 3.1, the definition of an interval, i.e., X = [sx, ex] implies ‘sx < ex’, and a set of basic *PIL axioms* [Zaidi & Wagenals, 2006] are used to construct this 4-symbol string representation for all possible temporal relations, i.e., both basic and compound relations, in Pointisible logic. A complete list of temporal relations and the corresponding string representations is provided in Zaidi and Wagenhals (2006).

The PG representation of PIL statements helps the inference mechanism of PIL to construct the string representation for the pairs of intervals with unknown relations by performing an *undirected* search [Ishaque, 2006] in the PG constructed after unification and folding processes. The algorithm presented in Ishaque (2006) searches for an undirected path between a pair of nodes (points) in a PG by applying a variant of a depth-first search algorithm. This algorithm, while traversing a PG, constructs an expression with the help of quantitative information available on the edges and the nature of the edges, i.e., LT and LE types, to determine the *distance* between the two nodes (points). An inference for a PIL relation between two points is made by *evaluating* an *expression* returned by the search algorithm. An inference for a PIL relation between two intervals requires four such searches to be performed, one for each pair of start/end points. The resulting string representation is pattern matched with the strings of all possible relations to identify the corresponding atomic/compound PIL relation. The quantitative information from the edges collected by the algorithm helps identify quantitative relations between the points involved.
As an illustration of the inference mechanism of PIL, an inference on the relationship between the two intervals ‘Z’ and ‘X’, in Figure 3.4, can be performed as follows: The four searches performed on the last PG in Figure 3.4, to construct the 4-symbol string representation, return ‘> < > =’ as the output. This string when pattern matched with the list of all relations identifies ‘Z *f* X’ (or, Z *finishes* X) as the inferred relation.

As mentioned earlier, the search algorithm in PIL performs an undirected traversal of a PG to make inferences. The need for an undirected search arises from the presence of quantitative information in PGs, i.e., edge lengths. Figure 3.5 illustrates a situation where an algorithm that looks for directed paths between a pair of nodes will not be able to infer the temporal relationship between points ‘A’ and ‘B’, since there is no directed path either from ‘A’ to ‘B’ or from ‘B’ to ‘A’. A search from node ‘A’ to ‘B’, using the algorithm in Ishaque (2006) returns ‘5 – δ – 10’ as the output expression, where δ represents the unknown, non-zero length on LT edge between nodes ‘C’ and ‘D’, and the negative signs represent the backward orientation of the edges traversed from ‘A’ to ‘B’. The expression represents the distance from point A to point B on a time line. The expression when evaluated results in a negative value, i.e., ‘5 – δ – 10 < 0’, thus establishing the temporal relationship ‘B < A’ (or, B *before* A).

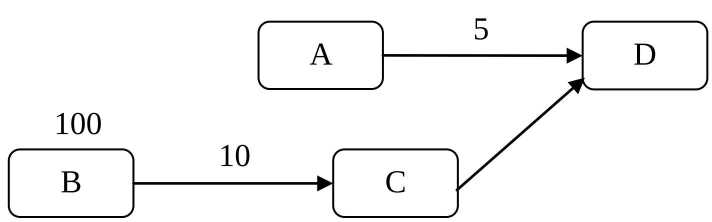


Figure 3.5: Inference in a PG

In addition to the inference algorithm used to infer temporal relations between a pair of points/intervals, a couple of similar algorithms are presented in Ishaque (2006) to calculate time stamps on points and lengths for intervals. The two algorithms try to identify an exact value for the stamp or the length. In case where exact value cannot be ascertained, the algorithms try to calculate an upper and/or lower bounds for the value. For example in Figure 3.5, a query for Stamp [D] is returned by the following: 'Stamp [D] > 110'.

The time complexity of the inference algorithms in PIL have a time complexity of $O(mn+n^2)$, where 'm' is the number of edges and 'n' is the number of nodes in a PG. The worst-case complexity, therefore, is $O(n^3)$ (in case of a complete graph when 'm' is to the order of n^2) with a much better average performance observed during empirical studies.

3.4 Deciding Consistency

The inference mechanism described above may result in erroneous and inconsistent results provided the system of PIL statements, represented by the PG, contains *inconsistent* information. The inference, on the other hand, is guaranteed to yield valid assertions given a consistent PIL system and corresponding PG representation. The following theorem characterizes inconsistency in PIL.

<u>*Theorem*</u>*: Inconsistency in PIL* [Zaidi & Wagenhals, 2006]

A system's description in PIL contains inconsistent information iff

(a) for some intervals X and Y, and atomic PIL relations Ri and Rj, both 'X Ri Y' and 'X Rj Y', $i \neq j$, or 'X Ri Y' and 'Y Rj X' (with the exception of = relation) hold true;

or

(b) for some intervals and/or points, the system can determine two string representations such that at least one pair of the algebraic inequalities representing relationships between the corresponding points represents an inconsistency. Let the two string representations be 'abcd' and 'uvwx', where a, b, c, d, u, v, w, and x $\in \{<, =, >, \leq, \geq, ?\}$. One of the (un-ordered) pairs of corresponding inequalities, i.e.,

(a, u), (b, v), (c, w), or (d, x) $\in \{(<, =), (<, >), (<, \geq), (=, >), (>, \leq)\}$;

or

(c) for a point p1, the system calculates two different stamps;

or

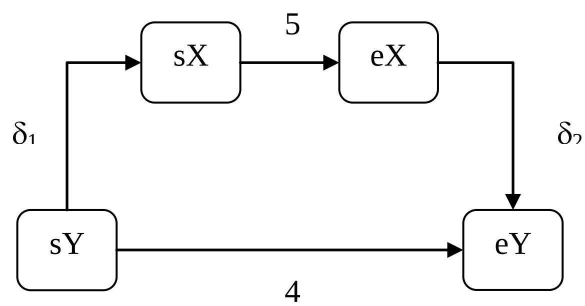
(d) for some points p1 and p2, 'p1 < p2', the system can determine two different lengths for the interval [p1, p2].

The verification mechanism of PG representation identifies these inconsistent cases by using a path-searching algorithm, [Ma, 1999]. The path-searching algorithm employs techniques by Busacker and Saaty (1965) and Warshall's algorithm (1962) to identify the erroneous cases. The inconsistencies identified in the theorem, above, manifest themselves in one of the two forms in the PG representation: (a) cycles, and (b) multiple paths between a pair of nodes in a PG with *infeasible* path lengths. An example of this second case is illustrated in Figure 3.6. The details of actual algorithms used to decide consistency in a PIL system can be found in Ishaque (2006).

Verification is the most costly step of the Point Graph construction process. It uses Warshall's algorithm for searching paths and has time complexity of $O(n^3)$, where n is the number of nodes

in a Point Graph. It must be noted that the $O(n^3)$ time complexity is only for detecting cycles and inconsistent paths in the Point Graph. Reporting all cycles or inconsistent paths can have an exponential time complexity, since there can be an exponential number of paths or cycles in a Point Graph, which may not be the case for most real-world PIL systems under study.

The total time complexity of Point Graph construction is $O(n^3)$, where $O(n^3)$ is due to the verification mechanism. But if it is known a priori that the set of PIL statements is consistent, the verification step can be avoided and the Point Graph can be constructed fairly quickly. This and several other techniques have been employed in the software implementation of the approach, called Temper, to make representation and reasoning algorithms more efficient. A brief introduction of Temper follows in the next subsection.



Infeasible Path Expressions:
$\delta_1 + 5 + \delta_2 = 4, \delta_1, \delta_2 > 0$

Figure 3.6: An Inconsistency in a PG

3.5 Temper

A software tool called Temper (<u>Temp</u>oral Programm<u>er</u>) implements the inference mechanism of Point-Interval Logic along with its verification and *revision* mechanisms. A screen shot of Temper's user interface is shown in Figure 3.7. Temper provides a language editor, shown in Figure 3.8, to input PIL statements and a query editor, shown in Figure 3.9, to run various queries on the constructed Point Graphs. It has a graphical interface to display the Point Graphs and also a text I/O interface to display information and results of the analyses (Figure 3.7). In the PG shown in Figure 3.7, each point is represented as a node, and each interval is represented by two nodes connected by a less than (<) or LT arc. Each LT edge is represented by a solid arc and the length, if available, appears adjacent to the arc. Each less-than-or-equal (≤) or LE edge is represented by a dotted arc. The stamp on each point appears inside the node representing the point. A special type of node, called virtual node, is used to represent *at least*, *at most*, *no later than*, *or no earlier than* temporal relations. The text callouts in Figure 3.7 identify the various elements of a Point Graph as displayed in Temper.

The implementation of PIL is in the form of a .NET class library called *PIL Engine* and provides an application programming interface (API) that can be used in any .NET compliant programming language. The tool Temper has been built using this API. It provides a graphical user interface to *PIL Engine*. It uses *QuickGraph*, which is an open-source C# implementation of *Boost Graph Library* [BGL], and *Graphviz* library from AT&T [Graphviz], for internal graph representation and for implementation of PIL algorithms.

The following section presents an application of Point-Interval Logic (PIL) to criminal forensics, especially in answering certain investigative questions relating to time-sensitive information about a terrorist activity. The illustrations are presented with the help of Temper generated outputs to the analysis. A set of temporal facts is taken from the internet on the London bombing incident that took place on July 7, 2005, as input to Temper's reasoning module. A hypothetical investigation on the information is carried out by constructing temporal queries in Temper to identify certain time intervals of potential interest to counter-terrorist investigators.

4. USING TEMPER FOR CRIMINAL FORENSICS – THE LONDON BOMBING

On July 7, 2005, there were four explosions in London at Tavistock Square, Edgware Road, Aldgate, and Russell Square. Three of these explosions, Edgware Road, Aldgate, and Russell Square, took place in trains that departed from King's Cross station. Images from close-circuit cameras installed at London's various railway stations were an important source of information for investigators. There were hours of images available from these cameras and the task of investigators was to analyze these images to identify possible suspects. The large number of such images, although desirable, can make an investigation that requires searching through them in a timely manner very time consuming. Time pressure was also created because there was need to identify the perpetrators quickly enough to apprehend any ones that survived.

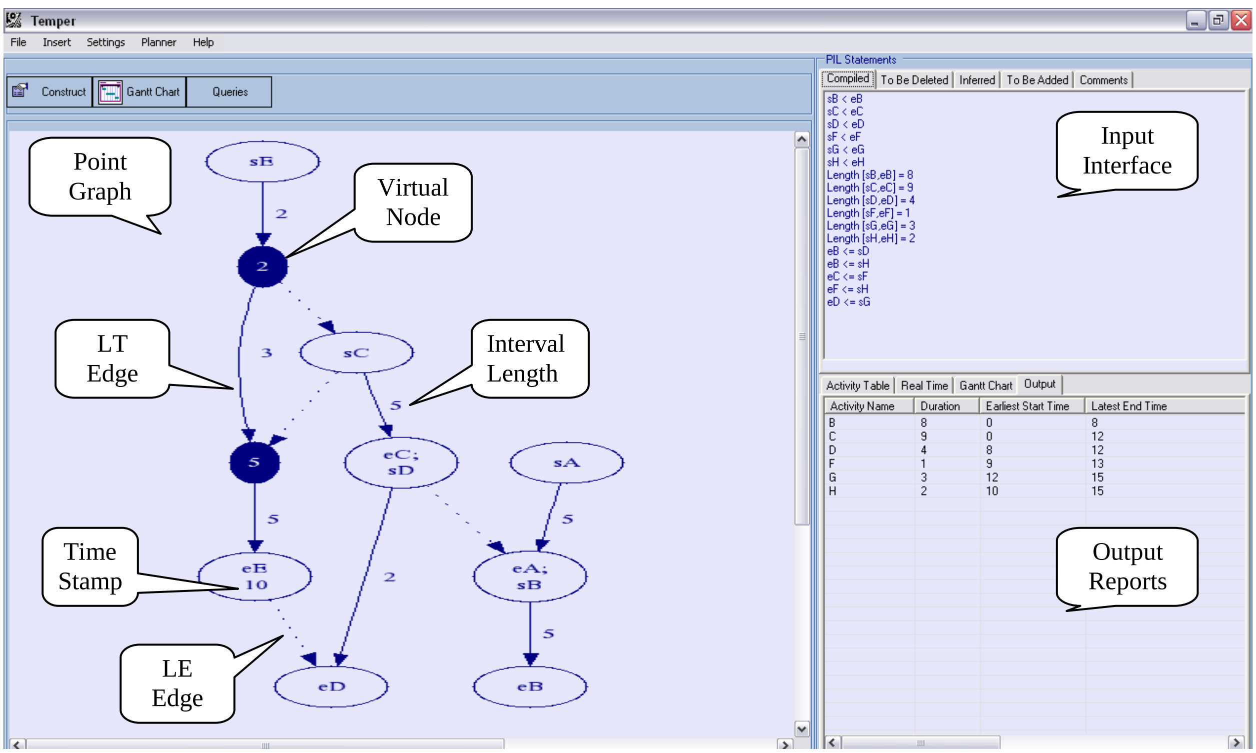


Figure 3.7: Temper User Interface with a PG Visualization

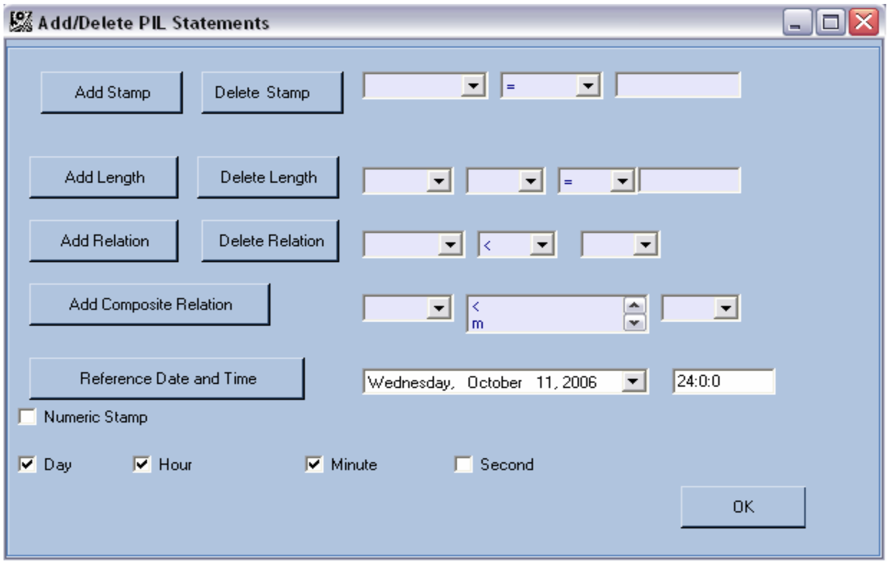


Figure 3.8: Temper's Language Editor

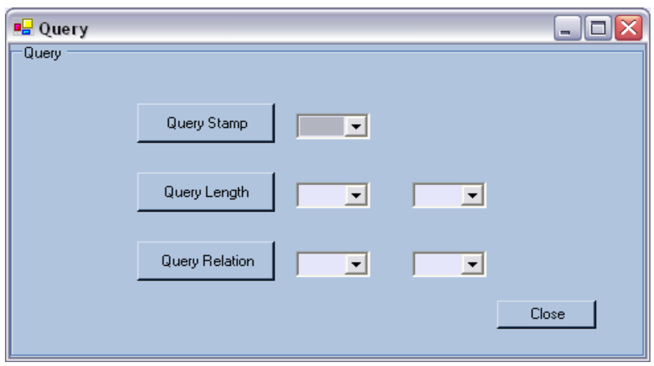


Figure 3.9: Temper's Query Editor

In this section, we demonstrate how temporal reasoning, in general, and especially Temper can be used to restrict the size of a potential interval for which to analyze images (by making sense of the available temporal information.) and thus speeding up the investigation. Since Temper has the ability to handle both qualitative and quantitative constraints, both types of information regarding the incident and/or the surrounding events can be input to it. Temper also offers the additional advantage of the verification mechanism that can be invoked to check the consistency of the available temporal information. This can be very useful when temporal information may originate from multiple (and possibly unreliable) sources. In this example, we demonstrate the capabilities of Temper by modeling a set of temporal information items related to the incident and by trying to identify the exact time or the shortest possible interval during which one of the ill-fated trains left from King's Cross station for Edgware. The graphical visualization of the temporal relations shows clearly where additional data are needed to establish temporal relations that will facilitate developing responses to specific queries.

The journey of the three trains from King's Cross station can be represented as PIL intervals. The journey of these trains ended in explosions. We also know the lower bounds on the travel times of these trains after their departures from King's Cross station, based on the distances of the sites of the explosions from King's Cross station. The train from King's Cross to Edgware

must have traveled for at least 5 time units. Similarly trains to Aldgate and Russell Square must have traveled for 4 and 5 time units, respectively. The time units are defined by a mapping from actual clock times to an equivalent representation on a real number line. Table 4.1 shows how this information can be represented as PIL statements. These PIL statements are typed or read into Temper using its language editor. Figure 4.1 shows the corresponding Point Graph in Temper.

Once the temporal information has been inserted, Temper can be used to draw inferences about the event(s)s of interest, i.e., the instant when one of the trains left King's Cross station for Edgware. We run a query, using the query editor of Temper, for the time stamp of the point "sTrain_KingX_Edgware" which represents the departure of the train from King's Cross station to Edgware. Figure 4.2 shows the query in Temper. The inference algorithm in Temper returns a '?' or 'unknown' as the result of the query, since the temporal information available to the inference mechanism is not enough to draw any meaningful relationship for the temporal event under investigation.

For the illustrated case, Temper cannot infer anything about the stamp of the event based on the information provided so far. This creates the need for information pull. Suppose, further investigation reveals that the explosion near Edgware took place between time units 840 and 845 (the explosion is considered to be an instantaneous event so the range 840 to 845 does not represent duration but the uncertainty in determining the actual occurrence time). Similarly the explosions near Aldgate and Russell Square occurred between 845 and 850, and between 840 and 850 respectively. These times are not actual clock times, rather their equivalent representation obtained by mapping the clock times on a real number line. Table 4.2 shows how this information can be represented as PIL statements. These PIL statements are added to the initial temporal model to get the Point Graph of Figure 4.3.

Once again, the query for the time stamp of the point "sTrain_KingX_Edgware" is executed (Figure 4.2). This time, Temper is able to determine an upper bound for the stamp of the event: its inference algorithm returns 'Stamp [sTrain_KingX_Edgware] <= 847', i.e., the train from King's Cross to Edgware must have left no later than 847.

Table 4.1: PIL Statements for London Bombing Scenario

Temporal Information	PIL Statements
Train traveling from King's Cross to Edgware	*interval* Train_KingX_Edgware
Train traveling from King's Cross to Aldgate	*interval* Train_KingX_Aldgate
Train traveling from King's Cross to Russell Square	*interval* Train_KingX_Russell_Sq
Explosion at Edgware	*point* Explosion_Edgware
Explosion at Aldgate	*point* Explosion_Aldgate
Explosion at Russell Square	*point* Explosion_Russell_Sq
Explosion at Edgware ended the journey of train from King's Cross to Edgware	Explosion_Edgware *f* Train_KingX_Edgware
Explosion at Aldgate ended the journey of	Explosion_Aldgate *f* Train_KingX_Aldgate

train from King's Cross to Aldgate	
Explosion at Edgware ended the journey of train from King's Cross to Russell Square	Explosion_Russell_Sq *f* Train_KingX_Russell_Sq
Train from King's Cross to Edgware traveled at least for 5 time units	*Length* [Train_KingX_Edgware] ≥ 5
Train from King's Cross to Aldgate traveled at least for 4 time units	*Length* [Train_KingX_Aldgate] ≥ 4
Train from King's Cross to Russell Square traveled at least for 5 time units	*Length* [Train_KingX_Russell_Sq] ≥ 5

Table 4.2: Additional PIL Statements for London Bombing Scenario

Temporal Information	PIL Statements
Explosion at Edgware happened no earlier than 840	*Stamp* [Explosion_Edgware] ≥ 840
Explosion at Edgware happened no later than 852	*Stamp* [Explosion_ Edgware] ≤ 852
Explosion at Aldgate happened no earlier than 845	*Stamp* [Explosion_Aldgate] ≥ 845
Explosion at Aldgate happened no later than 850	*Stamp* [Explosion_Aldgate] ≤ 850
Explosion at Russell Sq. happened no earlier than 840	*Stamp* [Explosion_Russell_Sq] ≥ 840
Explosion at Russell Sq. happened no later than 850	*Stamp* [Explosion_Russell_Sq] ≤ 850

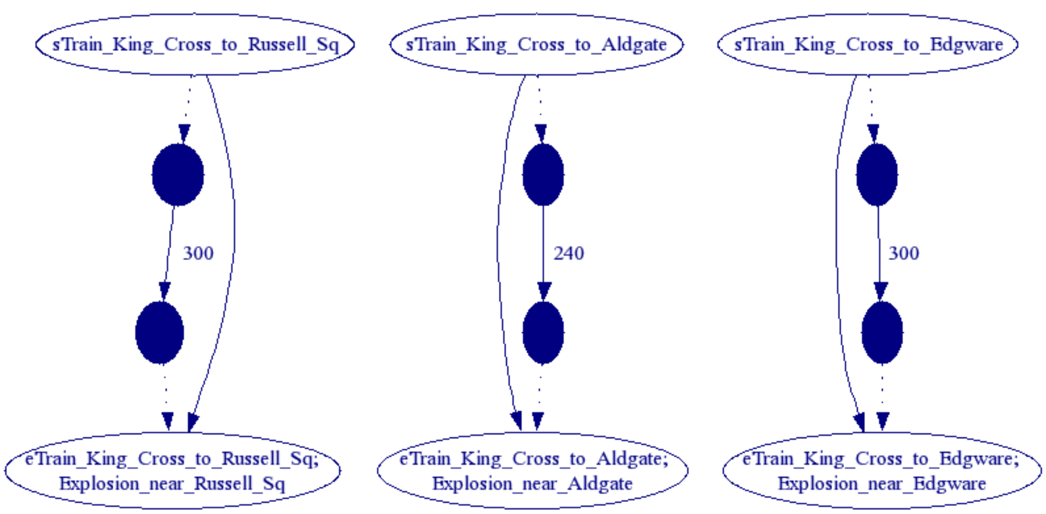


Figure 4.1: Point Graph for London Bombing Scenario in Table 4.1

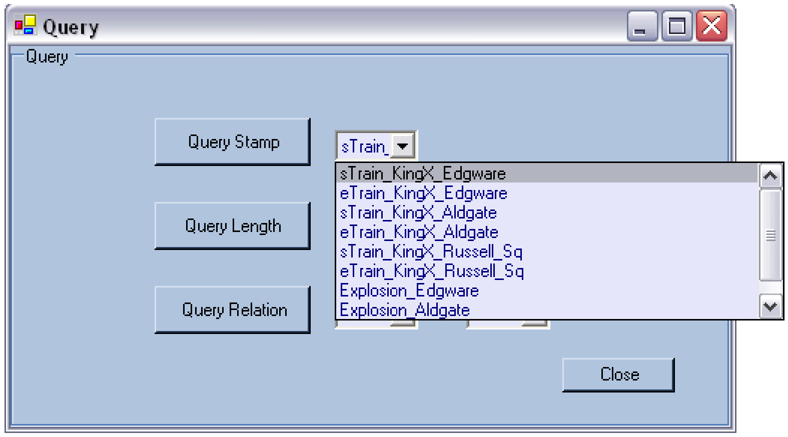


Figure 4.2: Running a Query in Temper

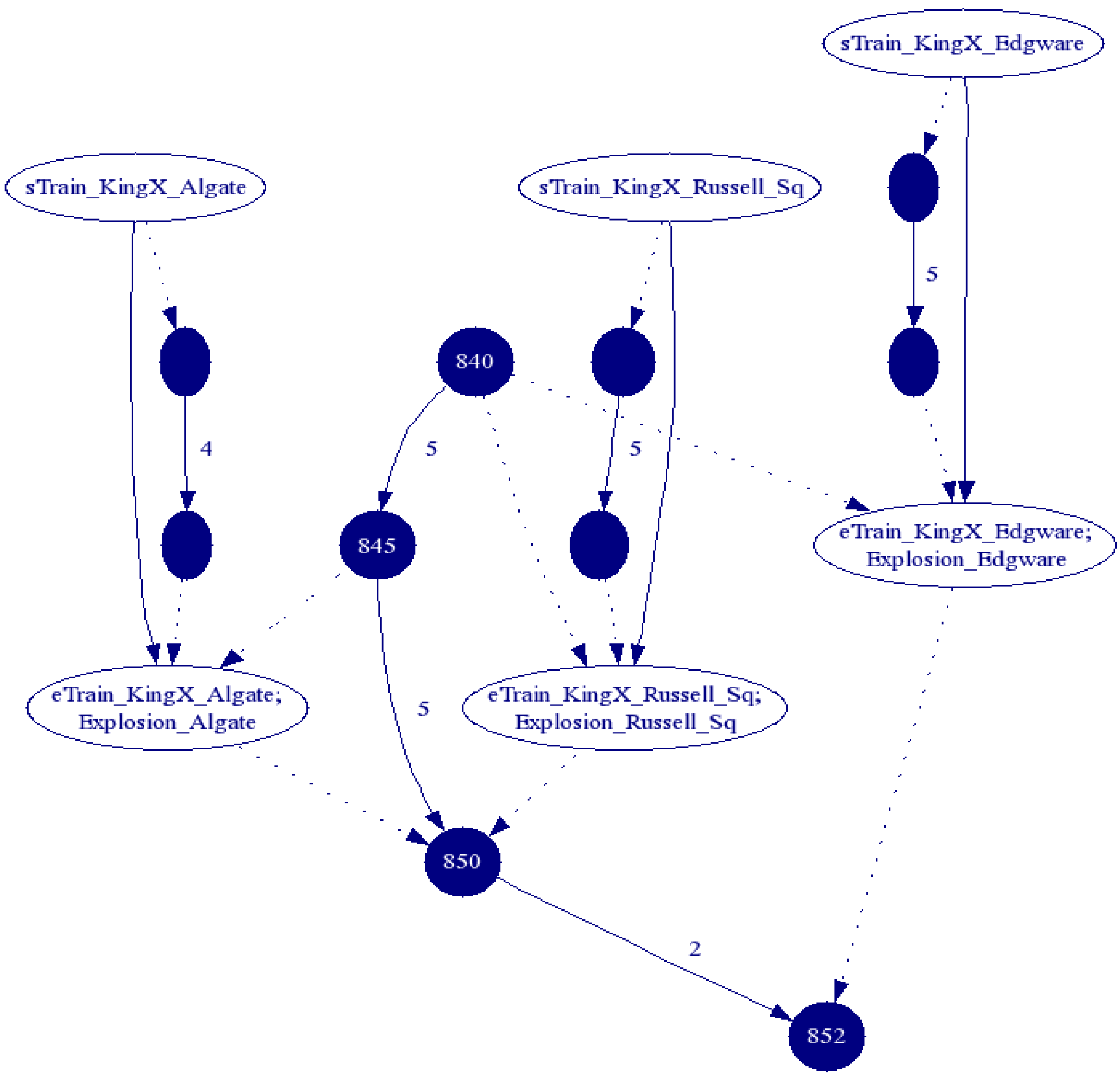


Figure 4.3. Revised Point Graph for London Bombing Scenario

As indicated earlier, the verification mechanism of Temper can detect the inconsistencies in the available temporal information. The ability to detect inconsistency can be very useful, especially when the information from different sources is combined into a single model of a situation under investigation. Suppose, we input to Temper the information that the train from King's Cross to Edgware left at time instant 848 (represented by the PIL statement: Stamp[sTrain_KingX_Edgware] = 848). Clearly, this statement is in conflict with the previously added PIL statements. Temper detects this inconsistency, and identifies the portion of the Point Graph that contains the contradiction (inconsistent paths in the Point Graph drawn with red color, also shown with an extra boundary around each node in the paths in Figure 4.4),. Note the two inconsistent paths (from node with stamp 840 to node with stamp 852): one path with length exactly equal to 12 units and the other path having a length of at least 13 units. We fix this inconsistency by deleting the last statement added, "Stamp [sTrain_KingX_Edgware] = 848". Temper's revision algorithm employs an efficient use of internal data structures to identify the portion of the Point Graph that is affected by the delete operation and only redraws that affected part to reconstruct the new PG representation. This saves a lot of computational effort required to reconstruct the Point Graph for an entire set of PIL statements from scratch every time there is a need to make modifications in the temporal information. A detailed description of the revision algorithm can be found in Rauf and Zaidi, (2002).

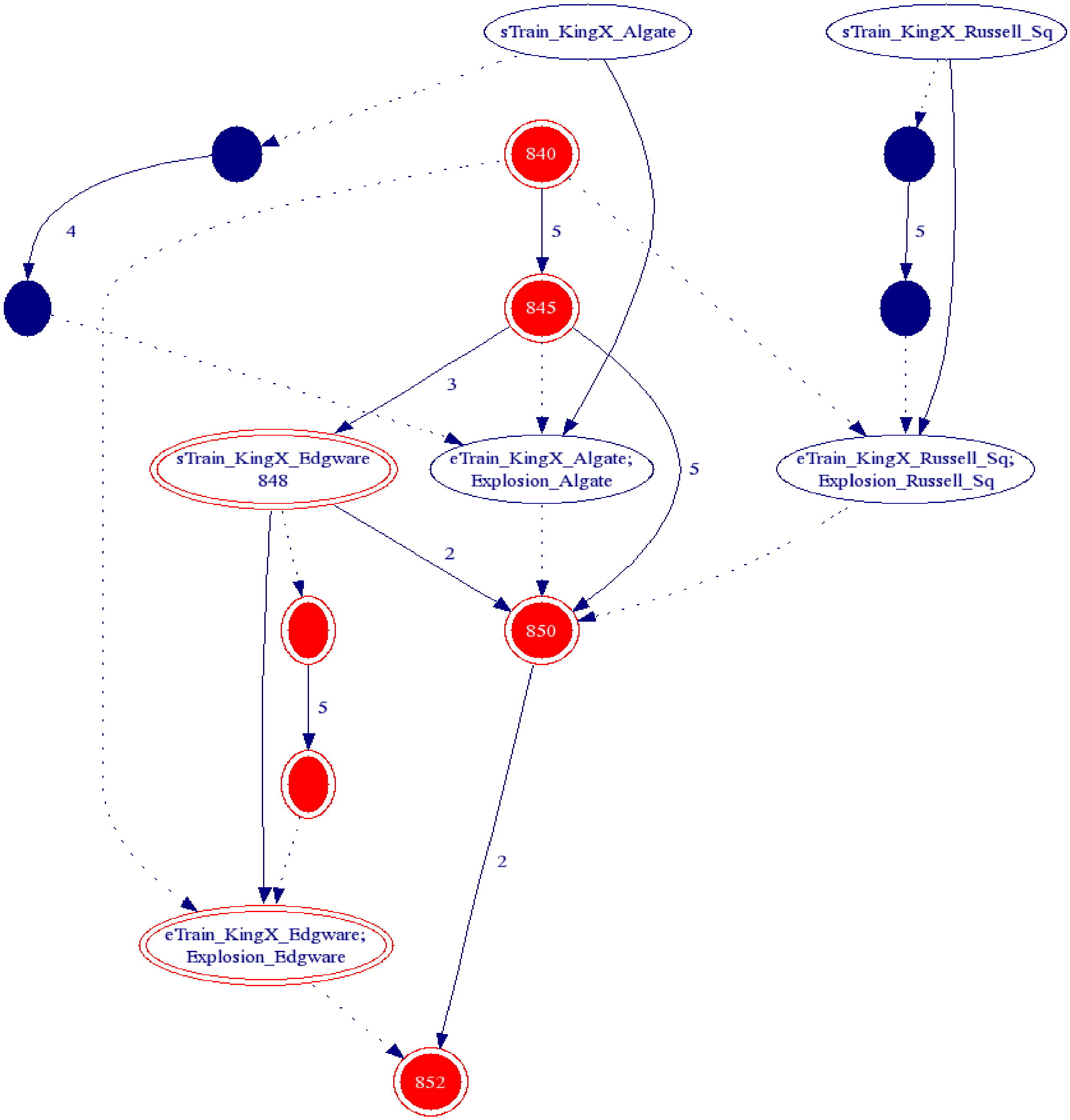


Figure 4.4: Inconsistency in the Point Graph

Suppose that the investigators have also identified four suspects who were spotted entering the Luton railway station at time instant 720. The investigators believe that these suspects took a train from Luton to King's Cross station, and at King's Cross station they boarded the trains in which the explosions took place. The next train from Luton to King's Cross departed at 748 and reached King's Cross at time instant 842. Obviously, if these suspects were in fact the bombers, the train from Luton should have reached King's Cross before any train in which there was an explosion left King's Cross station. This information can be represented by PIL statements as given in Table 4.3 and the resulting PG is shown in Figure 4.5. Note that Table 4.3 contains both qualitative and quantitative PIL statements.

The query for the time stamp of the point "sTrain_KingX_Edgware" is executed once more (Figure 4.2). This time, Temper is able to determine both an upper bound and a lower bound for the stamp of the event, its inference algorithm returns '842 < Stamp[sTrain_KingX_Edgware] <= 847', i.e., the train must have left King's Cross station after time instant 842 and no later than 847. Note that the lower bound is strict. Thus by applying the inference mechanism of Point-Interval Logic to the analysis of available temporal information, the approach has identified the bounds of the interval that we were interested in. The images need to be analyzed for this interval only; this improves the timeliness of the labor intensive image analysis process.

Table 4.3: Additional PIL Statements for London Bombing Scenario

Temporal Information	PIL Statements
Train traveling from Luton to King's Cross station	*interval* Train_Luton_KingX
Suspected spotted entering the Luton station	*point* Suspects_Spotted_at_Luton
Suspected spotted at Luton at time instant 720	*Stamp* [Suspects_Spotted_at_Luton] = 720
Train from Luton to King's Cross left at 748	*Stamp* [sTrain_Luton_KingX] = 748
Train from Luton to King's Cross arrived at 842	*Stamp* [eTrain_Luton_KingX] = 842
Train to Edgware left after the train from Luton	eTrain_Luton_KingX < Train_KingX_Edgware
Train to Aldgate left after the train from Luton	eTrain_Luton_KingX < Train_KingX_Aldgate
Train to Russell Sq. left after the train from Luton	eTrain_Luton_KingX < Train_KingX_Russell_Sq

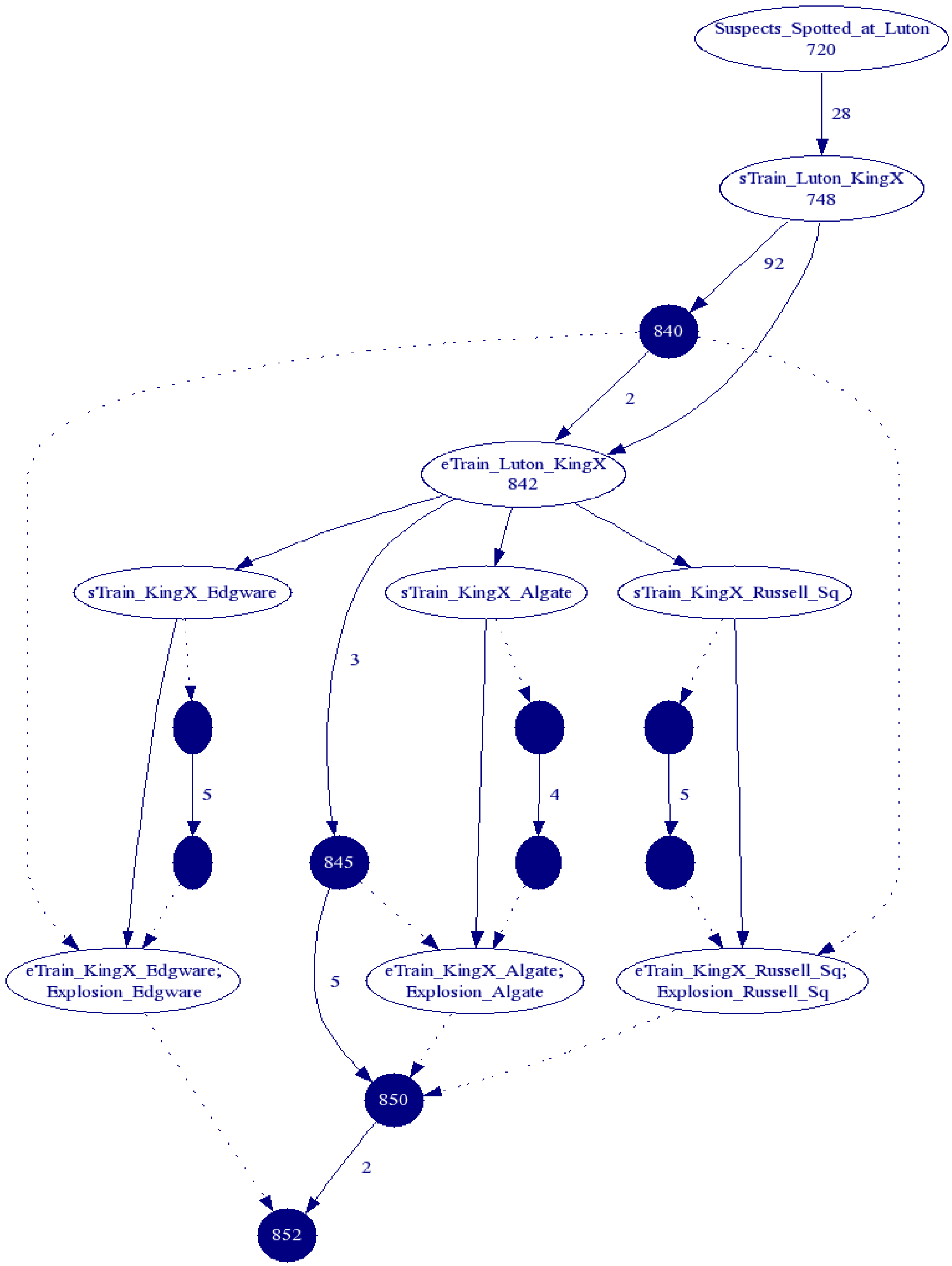


Figure 4.5: Revised Point Graph for London Bombing Scenario

5. CONCLUSION

This paper presented an illustration of the use of Point-Interval Logic (PIL) for creating temporal models of situations arising in forensics and for helping investigators answer relevant questions in a timely manner. The approach was demonstrated using Temper, which is a software implementation of Point-Interval Logic (PIL), and the London bombing incident as the scenario. The example presented demonstrates how Temper can be used in identifying a small interval for which the images from close-circuit cameras should be analyzed. The reader may argue that the problem could have been solved manually as well; that is true in the case of a small example like the one presented in this paper; however, in general situations the set of temporal statement may be too large for a human to handle. (The example set of PIL statements used in this paper is intentionally kept small for the sake of presenting the ideas and types of analyses that can be performed, instead of presenting the actual solution to the problem posed.) This presence of large set(s) of incomplete, inaccurate, and often inconsistent data calls for a computer-aided approach to such an analysis. Temper can combine temporal information from multiple sources, detect inconsistencies and identify the specific source(s) with inconsistent information. Temper can, therefore, be used to compare witness accounts of several individuals on the same incident for overlaps and inconsistencies—another useful application for forensics.

References

Allen, J. F. (1983). Maintaining Knowledge About Temporal Intervals, *Communications of ACM*, 26, (1983), 832-843.

Boost Graph Library. Information Available at: http://www.boost.org/

Busacker, R. G., and Saaty, T. L. (1965). Finite Graphs and Networks: an introduction with application. New York.: McGraw-Hill.

Drakengren, T. and Jonsson, P. (1997). Eight Maximal Tractable Subclasses of Allen's Algebra with Metric Time, *Journal of Artificial Intelligence Research*, 7, 25-45.

Findlay, J. N. (1941). Time: A treatment of some puzzles, *Aust. J. Phil.*, vol.19, pp. 216–235.

Gerevini, A. and Schubert, L. (1993). Efficient Temporal Reasoning through Timegraphs. In *Proceedings of IJCAI-93*.

Grahpviz. Information Available at: http://www.graphviz.org/

Ishaque, M.; Zaidi, A. K; and Levis, A. H. (2006). On Applying Point-Interval Logic to Criminal Forensics, *Proc. Of 2006 Command and Control Research and Technology Symposium (CCRTS).* San Diego, CA.

Ishaque, S. M. M. (2006). On Temporal Planning and Reasoning with Point Interval Logic, *MS Thesis*, CS, George Mason University, VA.

Kautz, H. (1999). Temporal Reasoning, In *The MIT Encyclopedia of Cognitive Science*, MIT Press, Cambridge.

Ladkin, P. B., and Maddux, R. D. (1988). Representation and reasoning with convex time intervals, *Tech. Report KES.U.88.2*, Kestrel Institute.

Ladkin, P. B. and Maddux, R. D. (1994). On binary constraint problems. *Journal of the Association for Computing Machinery*, 41(3):435-469.

Ma， C. (1999). On Planning Time Sensitive Operations, *MS Thesis*, SE, George Mason University, VA.

Prior, A. N. (1955). Diodoran modalities, *Phil. Quart*., vol. 5, pp. 205–213.

Rauf, I. and Zaidi, A. K. (2002). A Temporal Programmer for Revising Temporal Models of Discrete-Event Systems, in: *Proc. of 2002 IEEE International Conference on Systems, Man, and Cybernetics*, Hemmamat, Tunisia.

Vilain, M. and Kautz, H. (1986). Constraint-propagation algorithms for temporal reasoning. In *Proceedings of the Fifth National Conference on Artificial Intelligence,* pp. 377-382.

Vilain, M., Kautz, H., and van Beek, P. (1990). Constraint propagation algorithms for temporal reasoning: a revised report. In *Readings in Qualitative Reasoning about Physical Systems*, San Mateo, CA, Morgan Kaufman, 373-381.

Warshal,l S. (1962). A theorem on boolean matrices, *Journal of the ACM*, 9, 1, (1962), 11-12.

Zaidi, A. K. (1999). On Temporal Logic Programming Using Petri Nets. *IEEE Transactions on Systems, Man and Cybernetics*, Part A, 29(3): 245-254.

Zaidi, A. K., and Levis, A. H. (2001). TEMPER: A Temporal Programmer for Time-sensitive Control of Discrete-event Systems. *IEEE Transaction on Systems, Man, and Cybernetic,* 31(6):485-496.

Zaidi, A. K., and Wagenhals, L. W. (2006). Planning Temporal Events Using Point-Interval Logic. Special Issue of *Mathematical and Computer Modeling* (43), pp. 1229-1

10th International Command and Control Research and Technology Symposium
The Future of C2

1. Title: **Assessment of Effects Based Operations Using Temporal Logic***

2. Topic: Assessment, Tools, and Metrics/Effects Based Operations

3. Authors: **Abbas K. Zaidi** **Lee W. Wagenhals** **Sajjad Haider**

4. Point of Contact: **Abbas K. Zaidi**

5. Organization: System Architectures Lab, George Mason University

6. Address: System Architectures Laboratory
MSN 4B5
George Mason University
Fairfax, VA - 22030
703.993.1774 (v)
703.993.1706 (f)

7. Email: szaidi@ieee.org

Regular Paper

* This research was sponsored by the Air Force Research Laboratory, Information Directorate under Grant No. F30602-01-C-0065 and Air Force Office of Scientific Research under contract numbers F49620-01-1-0008.

Assessment of Effects Based Operations Using Temporal Logic[*]

Abbas K. Zaidi, Lee W. Wagenhals, Sajjad Haider
System Architectures Laboratory, MS 4B5
George Mason University
Fairfax, VA 22030
<szaidi2>,<lwagenha>,<shaider1>@gmu.edu

Abstract

This paper presents an application of temporal logic for assessment and analysis of courses of action (COA) in a Timed Influence Net (TIN) model for conducting Effects-Based Operations (EBO). The current practice in courses of action analysis looks at the impacts of actions on the likelihood of the desired effects over a period of time. The impact of time, however, is not studied explicitly. This paper illustrates the use of a point-interval temporal logic (PITL) for evaluating a COA's impact on desired objectives and undesired consequences both in terms of causal influences and timing of actions. The temporal formalism is also shown to run a what-if analysis for a better understanding of the temporal relationships between certain actions that may result in a desired effect at a particular time instant. The analysis, therefore, can be used for *fine-tuning* selected COAs for generating better plans. We return to the analysis of COAs that was reported in the 6th ICCRTS (Wagenhals, et al., 2001) to illustrate the use of this new technology. The hypothetical scenario that involves coalition operations to support Humanitarian Assistance to Indonesia is used for this illustration.

1. Introduction

Timed Influence Nets (TINs) have been used experimentally in the area of Effects-Based Operations (EBO) (Wagenhals and Levis 2002, Wagenhals et al. 2001 & 2003, Wentz and Wagenhals, 2004). They are used as a decision aid for modeling and analyzing uncertainties involved in a complex dynamic situation. Once a Timed Influence Net (TIN) model is constructed, it allows a system modeler to evaluate the performance of different courses of action in terms of their impacts on the likelihood of achieving some desired effect(s). The TIN formalism originates from a general class of probabilistic reasoning framework, known as Bayesian Networks, with the

[*] This research was sponsored by Office of Naval Research under Grant No. N00014-03-1-0033 and the Air Force Research Laboratory Information Directorate under Grant No. F30602-01-C-0065.

distinction that it integrates the notion of uncertainty and time into a single formalism (Wagenhals et al., 1998).

Temporal logic is a term used broadly to include approaches for the representation of temporal information within a logical and/or algebraic framework. A temporal logic can be defined as a language for encoding temporal knowledge about an application system and as a tool for reasoning about temporal relations among the system entities. Many different schemes have been suggested to represent time in the AI literature for both a qualitative and a quantitative treatment of time. A recent paper by Haider et al. (2005) explores the use of point-interval temporal logic PITL (Zaidi and Levis, 2001) for TIN models in analyzing the temporal impact of a certain course of action on variables of interest, i.e., objectives and/or undesired consequences. The inference mechanism of PITL is used to find, at a particular time instant, the source of a change, in terms of the actionable event, in the likelihood of a variable of interest. The PITL inference engine achieves this task by analyzing the relationships that exist between actionable events and the variable of interest. The analysis helps a system modeler in developing a better understating of the temporal relationships that must exist, at a particular time instant, between certain actions required to achieve a desired effect. This knowledge, in turn, can be used to sequence the actionable events on a time line so as to maximize the likelihood of the desired outcome.

This paper provides a brief background on TIN modeling and PITL's approach to temporal knowledge representation and reasoning. The paper then illustrates the use of the proposed temporal analysis of TIN models and the advantages it offers for the problem of planning, executing and assessing Effects-Based Operations (EBO). The potential of the analysis is examined and tested on some of the prototype systems presented earlier at Command and Control Research and Technology Symposia (Wagenhals and Levis 2002, Wagenhals et al. 2003, Wentz and Wagenhals, 2004).

The rest of the paper is organized as follows. Section 2 gives a brief definition of Time Influence Nets (TIN). A temporal language and a graph based representation for modeling it are presented in Section 3. Section 4 describes the temporal analysis of a TIN model that uses approach in Section 3. Section 5 illustrates the use of this new technology by applying it to a hypothetical, but realistic scenario. Section 6 provides conclusions.

2. Timed Influence Net

Timed Influence Nets (TIN) are used to model causal relationships between some desired effects and the set of actions that might impact their occurrence in the form of an acyclic graph. The actionable events in a TIN are drawn as root nodes (nodes without incoming edges). A desired

effect, or an objective in which a decision maker is interested, is modeled as a leaf node (node without outgoing edges). Typically, the root nodes are drawn as rectangles while the non-root nodes are drawn as rounded rectangles. Figure 1 shows a partially specified TIN. Nodes B and E represent the actionable events (root nodes) while node C represents the objective node (leaf node). The directed edge with an arrowhead between two nodes shows the parent node promoting the chances of a child node being true, while the roundhead edge shows the parent node inhibiting the chances of a child node being true. The first two elements in the inscription associated with each arc quantify the corresponding strengths of the influence of a parent node's state as being either true or false on its child node. The third element in the inscription depicts the time delay it takes for a parent node to influence a child node. For instance, event B, in Figure 1, influences the occurrence of event A after 5 time units.

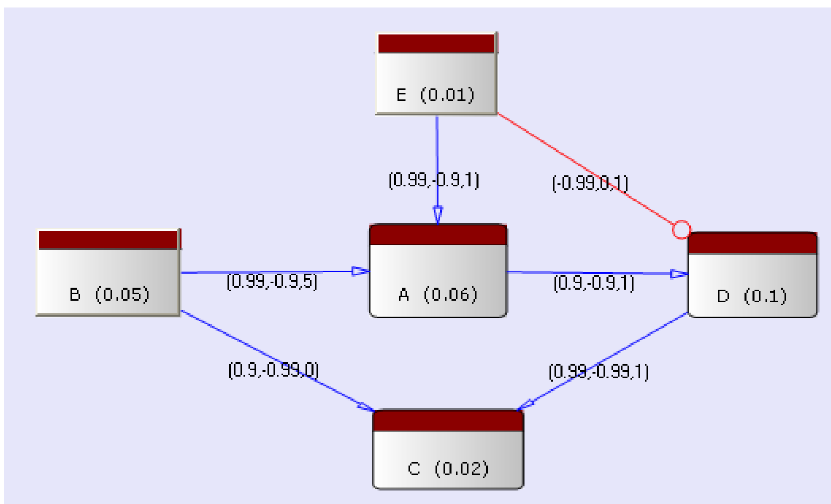


Figure 1. An Example Timed Influence Net (TIN)

The purpose of building a TIN is to evaluate and compare the performance of alternative courses of actions. The impact of a selected course of action on the desired effect is analyzed with the help of a *probability profile*. Consider the TIN shown in Figure 1. Suppose the following *input scenario* is decided: actions B and E are taken at times 1 and 7, respectively. Because of the propagation delay associated with each arc, the influences of these actions impact event C over a period of time. As a result, the probability of C changes at different time instants. A probability profile draws these probabilities against the corresponding time line. The probability profile of event C is shown in Figure 2.

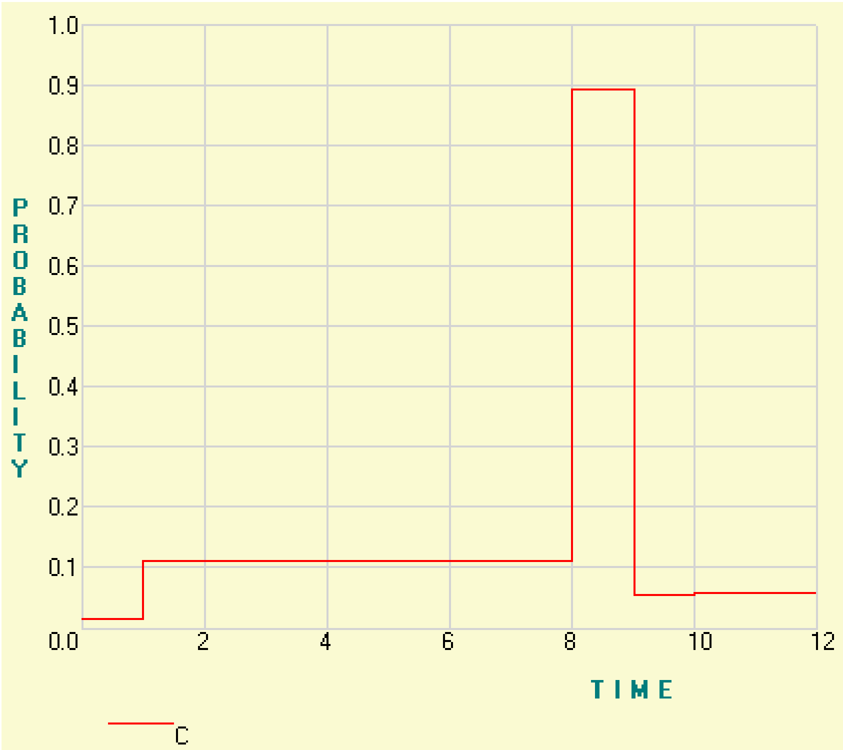


Figure 2. Probability Profile for Node C

The following items characterize a TIN:

1. A set of random variables that makes up the nodes of a TIN. All the variables in the TIN have binary states.
2. A set of directed links that connect pairs of nodes.
3. Each link has associated with it a pair of parameters that shows the causal strength of the link (usually denoted as g and h values). In Figure 1, the two parameters are the first two elements of the tuple associated with each arc as an inscription.
4. Each non-root node has an associated baseline probability, while a prior probability is associated with each root node (node without incoming edges). The baseline and the prior probabilities are shown as node inscriptions in Figure 1 right next to the node labels.
5. Each link has a corresponding delay d (where d $\geq$ 0) that represents the communication delay. In Figure 1, the node delays are the third elements on arc inscriptions.
6. Each node has a corresponding delay e (where e $\geq$ 0) that represents the information processing delay.
7. A pair (p, t) for each root node, where p is a list of real numbers representing probability values. For each probability value, a corresponding time interval is defined in t. In general, (p, t) is defined as:

$([p_1, p_2, \ldots, p_n], [[t_{11}, t_{12}], [t_{21}, t_{22}], \ldots, [t_{n1}, t_{n2}]])$,

where $t_{i1} < t_{i2}$ and $t_{ij} > 0 \; \forall \; i = 1, 2, \ldots, n$ and $j = 1, 2$

The set of all of the pairs of type defined in item #7, above, is referred to as input scenario, or sometimes (informally) as course of action. Analytically, a TIN can be described as:

A TIN is a tuple $(V, E, C, B, D_E, D_V, A)$ where

V: set of Nodes,

E: set of Edges,

C represents causal strengths: $E \rightarrow \{ (h, g)$ such that $-1 < h, g < 1 \}$,

B represents Baseline / Prior probability: $V \rightarrow [0, 1]$,

D_E represents Delays on Edges: $E \rightarrow Z^+$ (set of positive integers)

D_V represents Delays on Nodes: $V \rightarrow Z^+$ (set of positive integers), and

A: Input scenario. It represents the probabilities associated with the state of actions and the time associated with them.

A: $R \rightarrow \{([p_1, p_2, \ldots, p_n],[[t_{11},t_{12}], [t_{21},t_{22}], \ldots,[t_{n1},t_{n2}]])$, s.t. $p_i \in [0, 1]$, $t_{ij} \rightarrow Z^*$ and $t_{i1} < t_{i2}$, $\forall \; i = 1, 2, \ldots, n$ and $j = 1, 2 \}$

Where Z^* is the set of nonzero positive integers and R ($\subset$ V) represents the set of root nodes (actionable events).

3. Temporal Information and Point Graphs

A graph formalism called Point Graph (PG) is used in (Zaidi and Levis, 2001) to model temporal information. The approach is similar to *Precedence Graphs* with the added provision for quantitative temporal information. A node in a PG represents a time point and a directed arc between two nodes represents the temporal relation *Before* between the two time points. Formally, a Point Graph is defined as follows:

A Point Graph, PG (V, E_A, D, T) is a directed graph with:

V: Set of vertices with each node or vertex $v \in V$ representing a point on the real number line. Two points pX and pY are represented as a composite point [pX;pY] if both are mapped to a single point on the line.

E_A: Union of two sets of edges: $E_A = E \cup E_{\leq}$, where

E (LT edges): Set of edges with each edge $e_{12} \in E$, between two vertices v1 and v2, also denoted as (v1, v2), representing a relation '<' between the two vertices—(v1 < v2);

$E_{\leq}$ (LE edges): Set of edges with each edge $e_{12} \in E_{\leq}$, between two vertices v1 and v2, also denoted as (v1, v2), representing a relation '≤' between the two vertices—(v1 ≤ v2).

D (Length): Edge-length function (possibly partial):

$E \rightarrow Z^{+}$

T (Stamp): Vertex-stamp function (possibly partial):

$V \rightarrow Z^{*}$

The following temporal language can be used to describe temporal aspects/requirements of a system either already represented as a PG, or to be input to the PG representation.

The lexicon consists of the following primitive symbols:

Points (Event): A point X is represented as [pX, pX] or simply [pX]. Several labels p1, p2, …, pn, representing a single point are represented as a composite point [p1;p2;…;pn].

Intervals: An interval X is represented as [sX, eX], where 'sX' and 'eX' are the two end points of the interval, denoting the 'start' and 'end' of the interval, s.t. sX < eX.

Point Relations: These are the relations that can exist between two points. The set of relations R_P is given as: R_P = {Before, Equals, Procedes}

Functions: Interval length function that assigns a non-zero positive integer to a system interval, e.g.,

Length X = d, where X = [sX, eX], $d \in Z^{+}$

The stamp function assigns an integer number to a system point, e.g., Stamp p1 = t, $t \in Z$

A temporal statement in this language either takes the form of a function statement, or 'X Ri Y' where X and Y are points and $Ri \in R_P$.

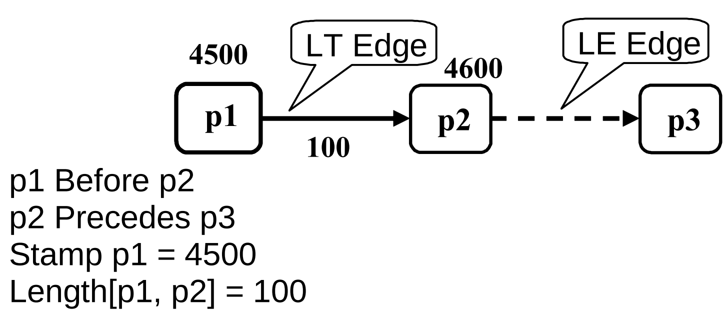


Figure 3. Point Graph and Corresponding Temporal Statements

The temporal relation 'Before' corresponds to the '<' edge in the PG definition. Similarly, the relation 'Precedes' corresponds to a '≤' edge, and the temporal relation 'Equals' results in a composite point (vertex) in the PG representation. The two functions for the quantitative

information directly map to the identically named functions in the PG definition. Figure 3 shows the correspondence

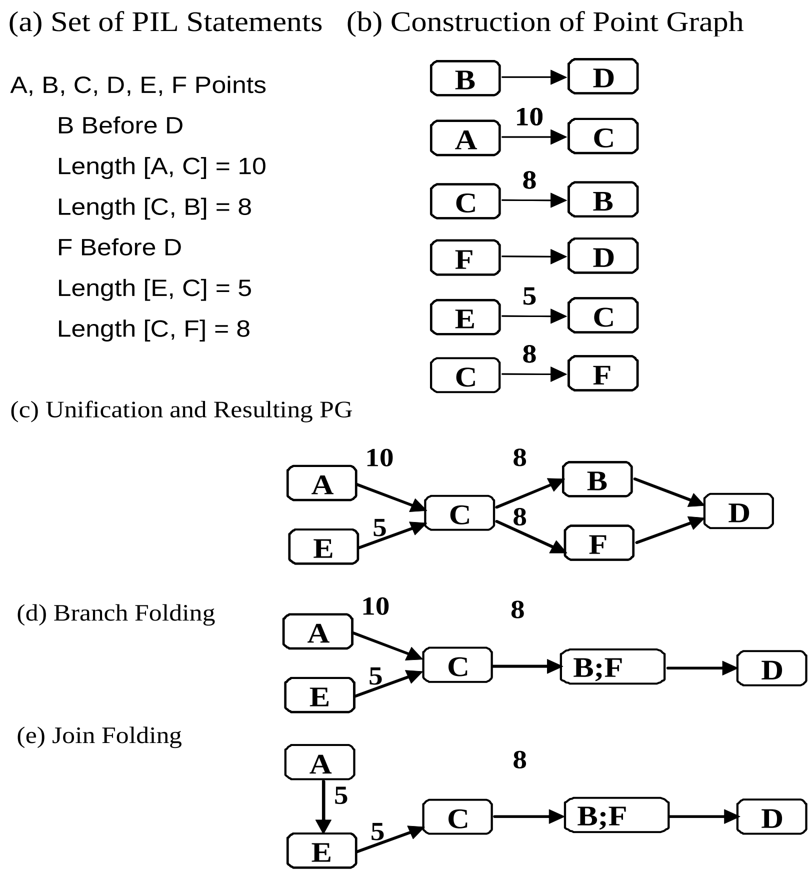


Figure 4. Steps in PG Construction

A set of PITL statements can now be represented as a set of PGs where each PG corresponds to a single statement in the temporal system. A consolidated PG for the entire temporal system can be constructed by *unifying* and *folding* the individual PGs (Zaidi and Wagenhals, 2004). The unification looks at the nodes of a set of PGs and merges the nodes with identical node labels or the ones with equality relation between them. The folding process, on the other hand, looks at the quantitative information on nodes, and edges, of a PG and folds the edges based on the available information. The process establishes new relations among system points, inferred through the quantitative analysis of the known relations specified by interval lengths and stamps. Figure 4 illustrates the process of constructing a PG for a set of PITL statements with the help of an example. An inference algorithm only needs to establish the presence or absence of paths between a pair of points to establish a temporal relation between the two. The lengths on the path help establish the exact, minimum, or maximum distance between two points in a query. For instance, in Figure 4, the

processing of the PG results in an edge between points A and E with a length of 5. The relation 'A Before E' with 'Length [A, E] = 5' is, therefore, the inferred temporal information.

A polynomial-time *path-consistency* algorithm is presented in Zaidi and Wagenhals (2004) for deciding the consistency of temporal information in the PG representation.

4. Temporal Analysis of Timed Influence Nets

Haider et al. (2005) present an approach for extracting the temporal information in a TIN and model it using the PG representation. The following is a brief description of the process of constructing a PG from temporal information in a TIN. Once a PG is constructed, the PITL approach is shown to help analyze temporal relationships among system variables in a complex uncertain situation. The results of this analysis can aid a system modeler in gaining a better insight of the impact of a selected course of action on desired effect(s). The PG representation of a corresponding TIN answers queries regarding certain temporal characteristics of an effect's probability profile. The PG also aids a system modeler by explaining what needs to be done for achieving a certain effect at a specific time instant. If the requirements for achieving effects at certain time instants are not temporally consistent, then the PG helps in understanding the reasons for inconsistencies.

4.1. Creating a Point Graph from a Timed Influence Net

The steps involved in generating a PG from a corresponding TIN are presented in Table 1. (Haider et al., 2005) The example TIN in Figure 1 is considered for an illustration of the algorithm. For this example case, R = {B, E}, F = {C}, and the input scenario is given as: 'B occurs at time 1 and E occurs at 7.' The following are the PITL statements that are extracted for the TIN as part of Step 2 of the algorithm.

Length [B1, A1] = 5	Length [A1, D1] = 1
Length [D1, C1] = 1	Length [E3, A3] = 1
Length [A3, D3] = 1	Length [D3, C3] = 1
Length [E4, D4] = 1	Length [D4, C4] = 1
B2 Equals C2	B1 Equals B2
E3 Equals E4	

The following PITL statements model the input scenario:

Stamp B1 = 1	Stamp E3 = 7

The PGs obtained as a result of Steps 2-5, in the algorithm, are shown in Figure 5.

Table 1. PG Construction from a TIN

Given a TIN

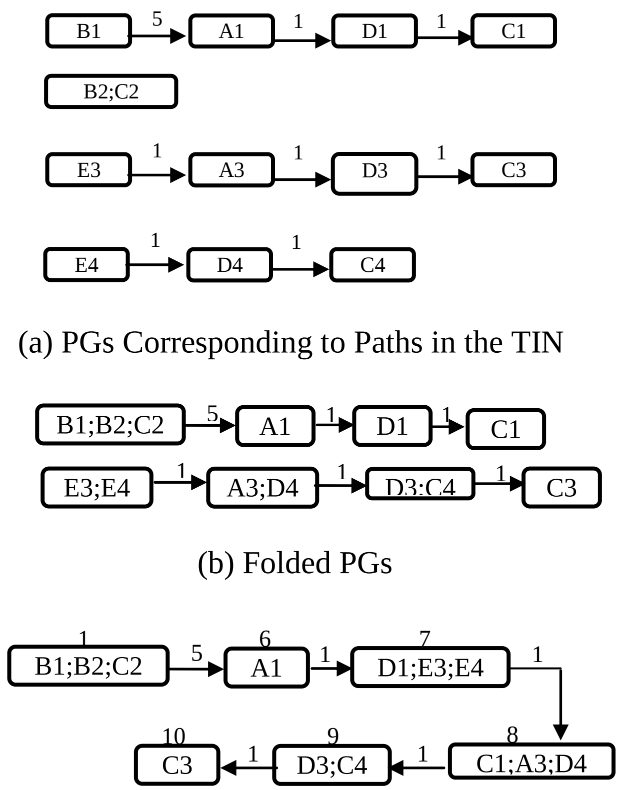
R: Set of Root Nodes (Actionable Events; nodes without incoming edges.)

F: Set of Leaf Nodes (Desired Effects; nodes without outgoing edges.)

1. For each $r \in R$ find all the paths leading to an $f \in F$. Apply this step to all $f \in F$.
2. Add a unique subscript to each node label in an individual path obtained in Step 1.
3. Represent each path as a PG where a node in the path becomes a vertex and a delay d (d >0) on an arc between two vertices v1, v2 becomes Length [v1,v2] = d in the PG.
4. For each set of vertices in PG that represent a root node in TIN, add a temporal equality relation 'Equal' among its elements.

The following step is executed once an input scenario is provided:

5. Based on the input scenario, assign time stamps to vertices representing root nodes.
6. Construct aggregate PG using temporal statements provided in Steps 3-5 after applying the unification and folding operations.



(a) PGs Corresponding to Paths in the TIN

(b) Folded PGs

(c) Final Folded and Unified PG with Input Scenario

Figure 5. Steps in Constructing the PG Representation for the TIN in Figure 1

4.2 Temporal Queries

Once a PG is obtained from a TIN, it can then be used to explain certain temporal characteristics of a probability profile. Consider the profile shown in Figure 2. Suppose a system modeler is interested in knowing what causes a change in the probability of event C at time 8. The algorithm that answers this and similar queries is presented in Table 2. Figure 6 shows the temporal query interface of *Pythia*[*]. The figure shows the query and the response obtained by the system by performing an inference on the underlying PG representation, shown in Figure 5. The response, as shown in the figure, states that the change in the profile of C at time 8 is because of action B occurring at time 1. Furthermore, if multiple paths exist between an action node and a desired effect, in a TIN, the algorithm of Table 2 can be used to identify the path through which the action has impacted the effect node. For the example under consideration, the path through which B impacted C at time 8 is: B – A – D – C. The Pythia interface is designed to highlight such paths for a visual check on the influences and their impacts on intermediate and end nodes.

Table 2: Answering Temporal Queries using a PG

Given a PG, a TIN, v: variable of interest, t: time of interest, C: list of Causes
1. Initialize C to null.
2. Determine the subscripts of v at time t. Let S = [s1, s2, …, sn] be the list of subscripts.
3. For each element s in S:
(i) Starting from the root of the PG, search the PG until the first variable with the subscript s is identified. Let x be such a variable.
(ii) Let m be the time stamp associated with x.
(ii) Add (x, m) to C.
4. Report the list C.

4.3 What-If Analysis

The PG obtained in Section 4.1 (Figure 5b) can also aid in performing what-if analyses. Suppose after observing the probability profile of Figure 2, the system modeler is interested in knowing what needs to be done, in order to combine the impact that reach C at time 8 with the impact that reach at time 9. Such a requirement may arise from a desire to cancel out the rise in probability at time 8 with the fall in probability at time 9, thus avoiding a potential window of high likelihood of the effect C in the interval from times 8 thru 9. The algorithm that accomplishes this task is presented in

[*] Pythia is a software application developed at the System Architectures Lab for constructing TIN models for planning and assessment of Effects-Based Operations. It is an advanced version of an earlier tool called CAESAR II/EB.

Table 3. Figure 7 shows the user interface for processing the What-If condition. The figure also shows the result obtained by the system with the help of PG in Figure 8. Since the length between points representing events B and E is 5 time units, to combine the impacts that affect node C at times 8 and 9, B must be executed 5 time units before E. This is, therefore, a necessary condition for the effect to materialize; however it might not be sufficient. The result of the analysis can be used to construct an input scenario that can then be tested on the TIN model to check if it really produces the desired effect. For the example under consideration, a possible input scenario suggested by the What-if condition can be described as: 'B occurs at time 1 and E occurs at 6.' As PITL statements, the scenario will be: Stamp B1 = 1 and Stamp E3 = 6. The resulting probability profile for node C is shown in Figure 9. The profile shows that the new scenario has successfully purged the high probability window present in the profile of Figure 2. The example illustrates how a system user can use the temporal analysis to acquire a better understanding of the impacts of actionable events on effects and then run a What-if analysis to identify temporal sequencing of these events to get the desired effects.

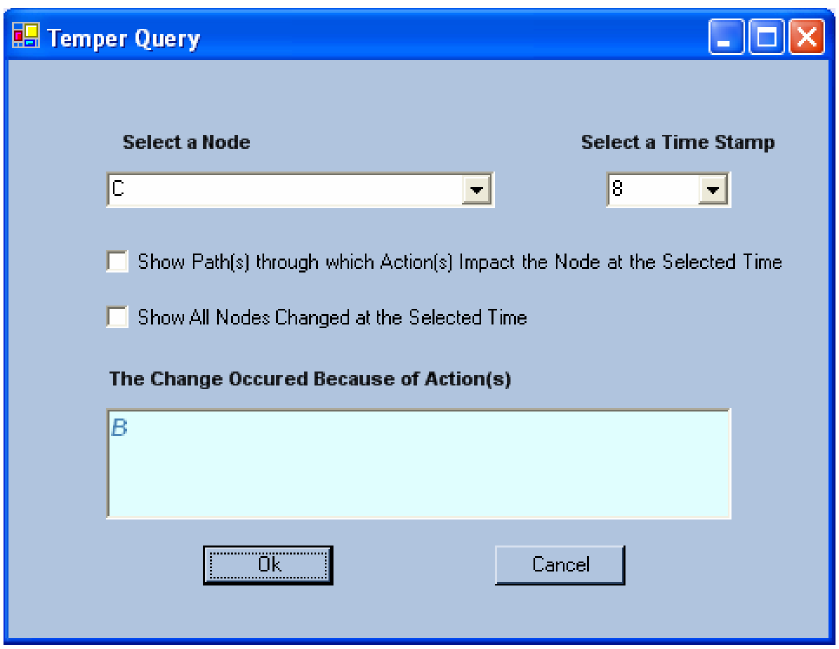


Figure 6. Query Processor Interface

The what-if analysis not only identifies the temporal relationships that should exist between two actionable events for achieving a desired impact at a certain time instant, but it also tells a system modeler if a given set of requirements are temporally inconsistent/infeasible.

Table 3. What-If Analysis Using a PG

Given a TIN, a PG G1

R: Set of Root nodes (Actionable Events) F: Set of Leaf nodes (Desired Effects)

V: List of variables of interest T: List of times of interest

S: List of variables with equal time stamp

1. For each r ∈ R find all the paths leading to an f ∈ F. Apply this step for all f ∈ F.
2. Add a unique subscript to each node in an individual path obtained in Step 1.
3. Represent each path as a PG where a node in the path becomes a vertex and a delay d (d >0) on an arc between two vertices v1, v2 becomes Length(v1,v2)=d in the PG.
4. For each set of vertices in PG that represent a root node in TIN, add a temporal equality constraint ‘Equal’ among its elements.
5. For each element v in V,

   (i) Find its subscript at the corresponding time t (t ∈ T) in G1. Let s be the subscript.

   (ii) Add variable v with subscript s to S.
6. Add temporal equality relation ‘Equals’ among the elements of list S.
7. Construct aggregate PG G2 using temporal statements provided in Steps 3-6.

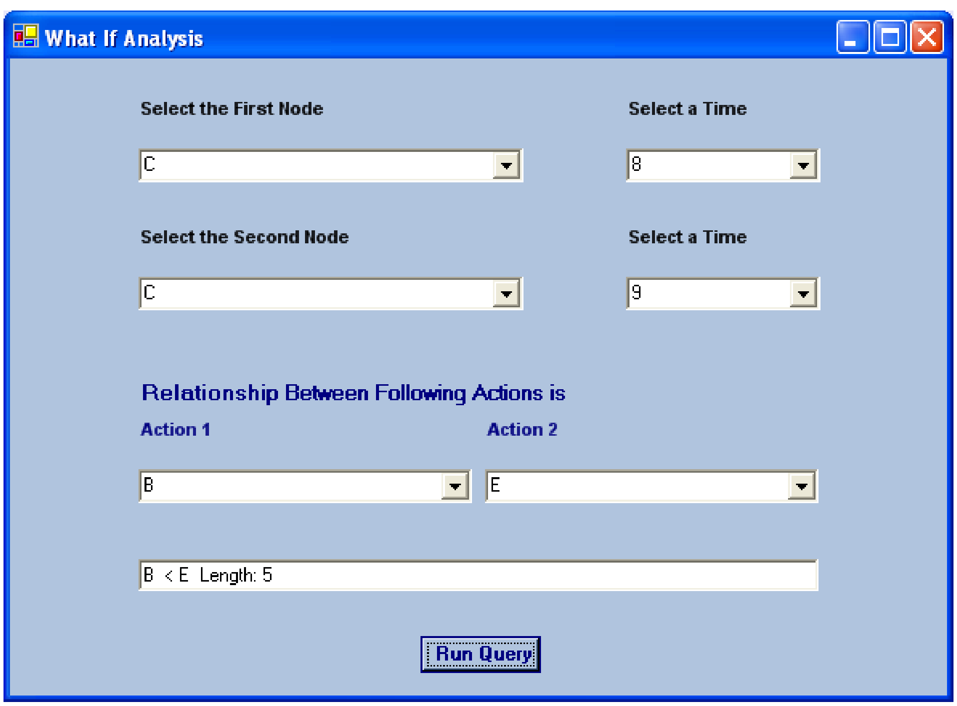


Figure 7. Illustration of What-If Analysis

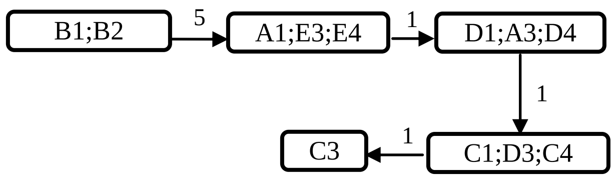


Figure 8. PG Used for What-If Analysis

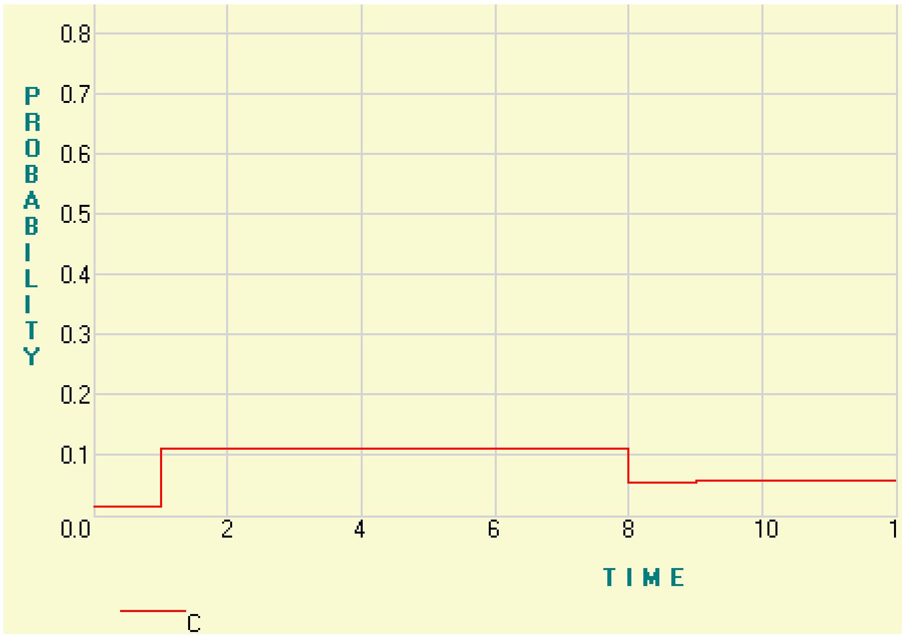


Figure 9. Probability Profile for New Scenario

5. Application

A scenario that was plausible but fictitious was developed for the purpose of demonstrating the potential use of the TIN and PITL combined capabilities for COA analysis. The scenario is the same one that was presented in the 6th ICCRTS (Wagenhals, et al., 2001). In the scenario, internal political instabilities in Indonesia have deteriorated and ethnic tensions between the multiple groups that comprise Indonesia have increased. Religion has been a major factor in these conflicts. Members of one of the minority (2%) religious groups have banded together to combat disenfranchisement.

These members have formed a rebel militia group. Armed conflict recently occurred between these rebels and the Indonesian military. The rebels fled to eastern Java where they have secured an enclave of land. This has resulted in a large number of Indonesian citizens (estimates of about 10,000) who are within the rebel-secured territory. Many of these people are unsympathetic to the rebels and are considered to be at risk. It is feared that they may be used as hostages if ongoing negotiations break down with the Indonesian government. The food and water supply and sanitation facilities are very limited within the rebel-secured territory.

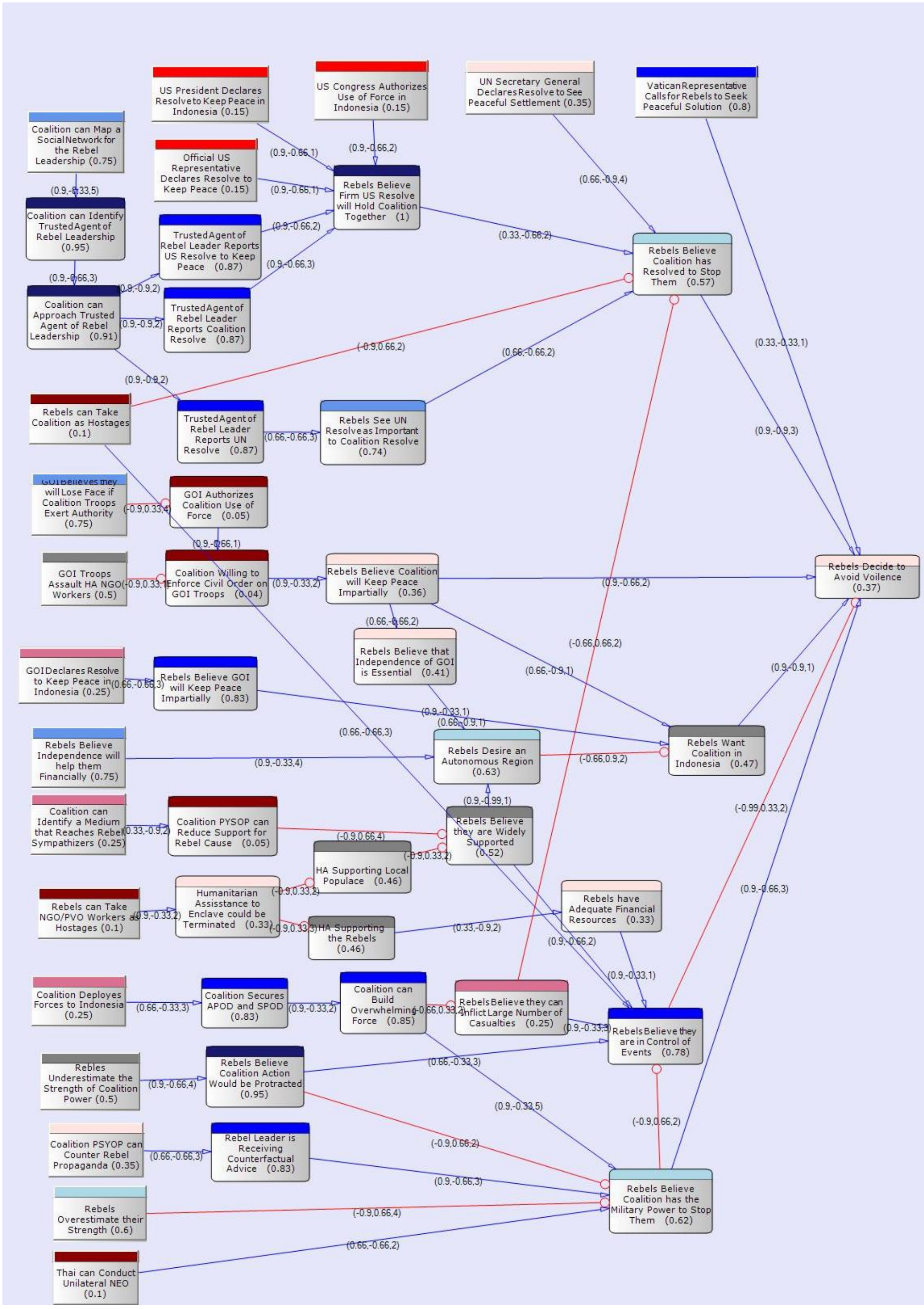


Figure 10. Timed Influence Net of East Timor Situation

Several humanitarian assistance (HA) organizations are on the island, having been involved with food distribution and the delivery of public health services to the urban poor for several years. So far, the rebels have not prevented HA personnel from entering the territory to take supplies to the citizens. The U.S. and Australian embassies in Jakarta are closely monitoring the situation for any indications of increasing rebel activity. In addition, Thailand, which has sent several hundred

citizens to staff numerous capital investment projects on Java, is known to be closely monitoring the situation.

A TIN has been created that reflects the situation and can be used to analyze potential COAs (see Figure 10).

The TIN models the causal and influencing relationships from actionable events (on the left side and along the top of the model in Figure 10) and the overall effect of concern which is the single node with no parents on the right-hand side of the model. In this case, the effect is "Rebels decide to avoid violence".

The actionable events in this model include a combination of potential coalition, UN, and rebel actions. The coalition actions include actions by the US government, its military instrument of national power, actions by the Government of Indonesia, and actions by Thailand.

For purposes of illustration, we have created a proposed course of action that contains potential actions by the coalition, UN, and the Rebels along with the timing of those actions. The TIN model was executed with that COA (COA #1) and the probability profile for the overall effect was generated as shown in Figure 11.

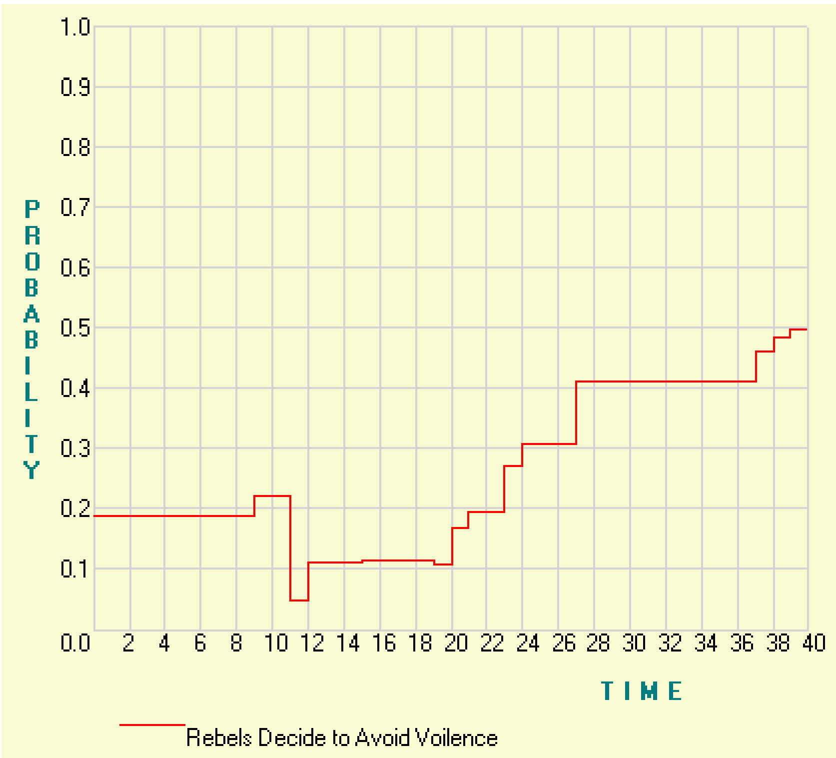


Figure 11. Probability Profile for Initial COA (COA #1)

It is noted that initially it is unlikely that the rebels will decide to avoid violence. In fact the projection is that matters will get worse before they start to improve. Of particular concern is the

drop in the probability profile at time 11 with a rise at time 12. The goal is to see if changing the COA can eliminate the "dip" shown between time 11 and 12. The question is what are the actions that are causing the changes in the probability profile and can the timing of those actions be changed to positively affect the probability profile.

To do this analysis the TIN is converted to a point graph and the temporal query function is used to address the cause for the decrease in the probability profile at time 11 and the increase at time 12. The two Pythia generated temporal query windows are shown in Figure 12.

The algorithm, presented in Section 4.2, shows that there are two actions that cause the downward change in the probability profile at time 11 ("Government of Indonesia Troops Assault Humanitarian Assistance Workers" and "Rebels take Hostages") and one action ("UN Secretary General Declares Resolve to See Peaceful Resolution") that causes the increase at time 12.

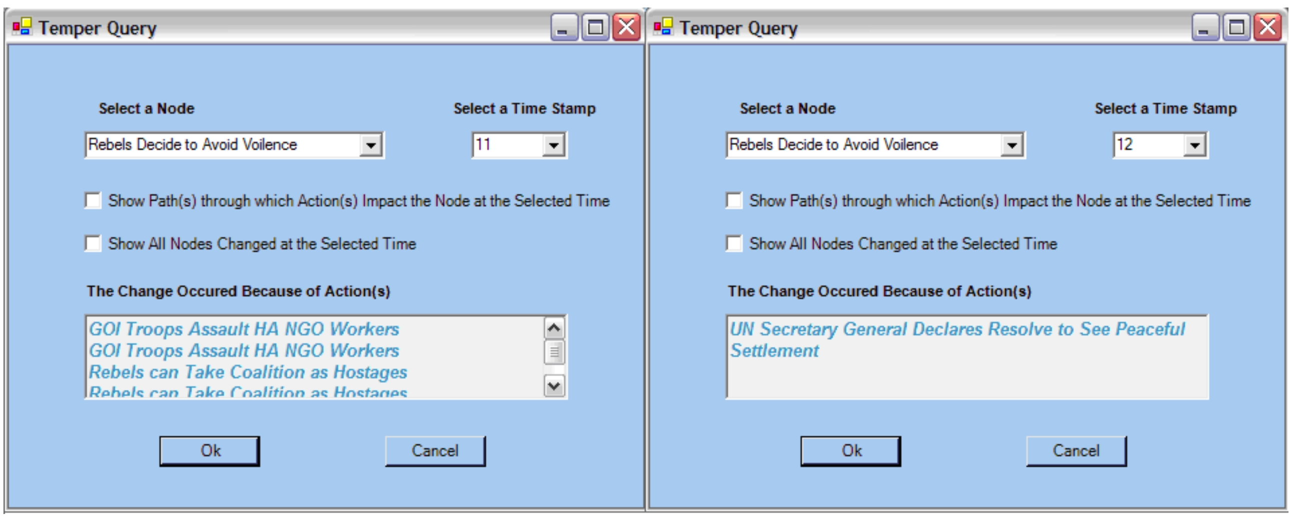


Figure 12. Temporal Queries on Actions that Cause the Change in the Probability Profile

It is believed that Rebels may take the Hostages at time 6, so a "What if" temporal query is made to see if asking the UN Secretary General to make his declaration at a different time can positively impact the assessment. The query is shown in Figure 13. In this query, we are looking at the same effect node (Rebels decide to avoid violence) at two different times (11 and 12) and seeing what the temporal relationship needs to be between the two actions under consideration to cause the changes at times 11 and 12 to occur at the same time. The assumption is that by making them occur at the same time, the positive change will counter the negative change. The result of the query says that the UN Secretary's action must occur at least two time units before the Rebel action. Since intelligence believes that the Rebel action might occur at time 6, the COA is changed to have the

UN Secretary make his declaration at time 4. The new COA is termed COA #2. A new probability profile is generated and compared to the original probability profile as shown in Figure 14.

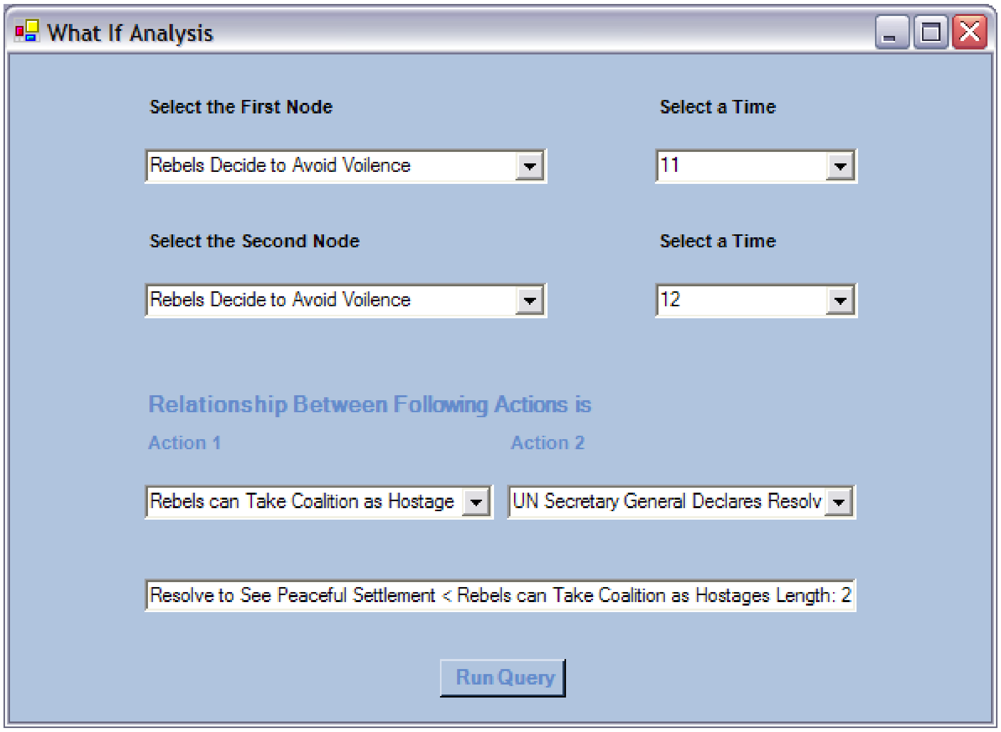


Figure 13. What if Analysis Query

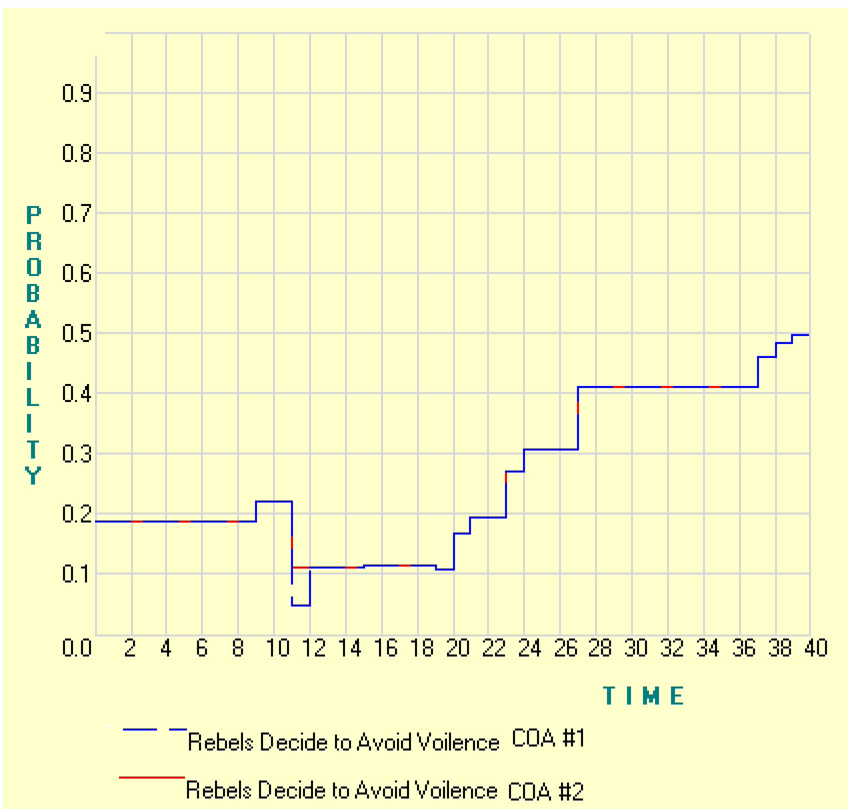


Figure 14. Probability Profile Comparison Based on Change in UN Secretary’s Action

The comparison of the profiles shows that the proposed action eliminates the "dip" in the probability profile between time 11 and 12.

We also note that the Government of Indonesia (GOI) assault actions also have a negative impact on the probability from the first temporal query of Figure 12. The TIN analysis shows that eliminating this action results in a further improvement in the probability profile as shown in Figure 15. The new scenario is shown as COA #3.

This analysis could provide the rationale to persuade both the UN and the GOI to modify their plans.

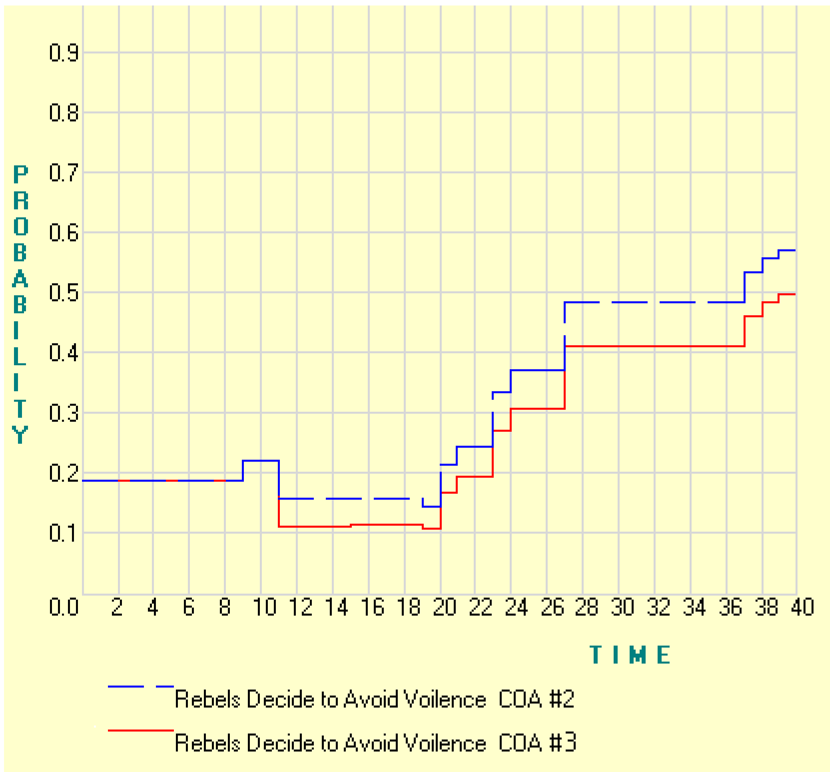


Figure 15. Improvement in Probability Profile if GOI Does Not Assault

6. Conclusions

Previous experience with building effects based TIN models has shown that the analysis of the COAs was very time consuming and difficult. This was in part because there was no easy way to determine which actions were causing changes in the probability profile or any way to determine the temporal relationships between actions and the changes in the probability profiles other than by trial and error. By combining the rigor of the point-interval temporal logic with the TIN representation, it has been shown to be possible to provide this capability to the COA analyst. Of course, the new capability still requires the analyst to determine which queries to make and requires the analyst to determine how the COA might be adjusted based on those queries. In the future it may be possible to provide a more automated algorithm that makes use of the PITL capability and

the features of the probability profile to assist the analyst in adjusting the COAs to improve the projections on achieving effects. This is a topic of on-going research.

References

Haider, S., Zaidi, A. K., and Levis, A. H. 2005. On Temporal Analysis of Timed Influence Nets using Point Graphs, *In proceedings of The 18th International FLAIRS Conference*, FL. (In Press.)

Wagenhals, L. M., Shin, I., and Levis, A. H. 1998. Creating Executable Models of Influence Nets with Colored Petri Nets. *Int. J. STTT*, No. 2, 168-181.

Wagenhals, L. W., T. J. Reid, R. J. Smillie, & A. H. Levis 2001. Course of Action Analysis for Coalition Operations. Proc. 6th International Command and Control Research and Technology Symposium, Annapolis, MD.

Wagenhals, L. W. and Levis, A. H. 2002. Modeling Support of Effect-Based Operations in War Games. *In Proceedings of the Command and Control Research and Technology Symposium*, Monterey, CA.

Wagenhals, L. W., Levis, A. H., and McCrabb, M. B. 2003. Effects-Based Operations: A Historical Perspective of a Way Ahead. *In Proceedings of the Eighth International Command and Control Research and Technology Symposium*, Washington DC.

Wentz, L. K. and Wagenhals, L. W. 2004. Effects Based Operations for Transnational Terrorist Organizations: Assessing Alternative Courses of Action to Mitigate Terrorist Threats. *In Proceedings of the Command and Control Research and Technology Symposium*, San Diego, CA.

Zaidi, A. K. and Levis, A. H. 2001. TEMPER: A Temporal Programmer For Time-Sensitive Control Of Discrete Event Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 31(6), 485-496.

Theory of Influence Networks

Abbas K. Zaidi, *Senior Member, IEEE*, Faisal Mansoor, P. Papantoni-Kazakos, *Fellow, IEEE*

***Abstract*— Influence Networks are Bayesian Networks whose probabilities are approximated via expert provided influence constants. They represent a modeling and analysis formalism for addressing complex decision problems. In this paper, we present a comprehensive theory of Influence Networks that incorporates design constraints for consistency, temporal issues and a dynamic programming evolution of the Influence Constants. We also include numerical evaluations for a specific Timed Influence Network.**

***Index Terms*—Bayesian Networks, Influence Networks, Timed Influence Networks**

I. Introduction

The easy access to domain-specific information and cost-effective availability of high computational power have changed the way people think about complex decision problems in almost all areas of application, ranging from financial markets to regional and global politics. These decision problems often require modeling of informal, uncertain, and unstructured domains in order for a decision maker to evaluate alternatives and available courses of actions. The past decade has witnessed an emergence of several modeling and analysis formalisms that try to address this need, the most popular one being represented by Probabilistic Belief Networks [19, 21], most commonly known as Bayesian Networks (BNs).

BNs model uncertain domains probabilistically, by presenting the network nodes as random variables. The arcs (or directed edges) in the network represent the direct dependency relationships between the random variables. The arrows on the edges depict the *direction* of the dependencies. The strengths of these dependencies are captured as conditional probabilities associated with the connected nodes in a network. A complete BN model requires specification of all conditional probabilities prior to its use. The number of conditional probabilities on a node in a BN grows exponentially with the number of inputs to the node, which presents a computational challenge, at times. A major problem in BNs may be the specification of the required conditional probabilities, especially when either objective values of these probabilities can not be provided by experts or there exist insufficient empirical data to allow for their reliable estimation.

Recognizing the problem in the specification of conditional probabilities in BNs, in conjunction with the computational complexity involved, Chang et al [2] developed a formalism, at George Mason University, named Causal Strength (CAST) logic, as a intuitive and approximate language. The logic utilizes a pair of parameter values to represent conditional dependency between two random variables, where these parameter values model assessed (by experts) mutual influences between events. The CAST logic approximates conditional probabilities via influence relationships and is used as a knowledge elicitation interface to a underlying BN.

The CAST logic approach was later extended to represent relationships between events involved in network interconnections, as in BNs. The extension is basically a BN with conditional probabilities approximated via the use of influence parameters and was named Influence Nets (INs) [22, 23-25]. INs require an expert who specifies the influence parameter values and their interrelationships, as well as some a priori probabilities, all needed for the approximation of the pertinent conditional probabilities. As basically modified BNs, the objective of INs is to compute the probabilities of occurrence of sequential dependent events, and do not provide recommendations for actions. However, the probabilities of occurrence computed by the INs may be utilized by activation networks towards the evaluation and recommendation of actions [20].

Manuscript received March 01, 2007. This work was supported by the Air Force Office of Scientific Research (AFOSR) under Grants FA9550-05-1-0106 and FA9550-05-1-0388.

Abbas K. Zaidi is with George Mason University, Fairfax, VA 22030, USA. (Corresponding author, Ph: 703-993-1774; fax: 703-993-1601; e-mail: szaidi@ieee.org).

Faisal Mansoor is a graduate student at George Mason University, Fairfax, VA 22030, USA. (email: fmansoor@gmu.edu).

P Papantoni-Kazakos is with the University of Colorado, Denver, CO 80202, USA. She is currently on sabbatical leave and is associated with the System Architectures Lab at George Mason University as a Visiting Professor. (email: tpapanto@gmu.edu).

BNs and INs are designed to capture *static* interdependencies among variables in a system. A situation where the impact of a variable takes some *time* to reach the affected variable(s) cannot be modeled by either one. In the last several years, efforts have been made to integrate the notion of time and uncertainty. Wagenhals et al. [27, 28, 32] have added a special set of temporal constructs to the basic formalism of INs. The INs with these additional temporal constructs are called Timed Influence Nets (TINs). TINs have been experimentally used in the area of Effects Based Operations (EBOs) for evaluating alternate courses of actions and their effectiveness to mission objectives in a variety of domains, e.g., war games [29, 30, 31, 34], and coalition peace operations [33], to name a few. The provision of time allows for the construction of alternate courses of action as timed sequences of actions or actionable events represented by nodes in a TIN [29, 28, 34]. A number of analysis tools have been developed over the years for TIN models, to help an analyst update beliefs [5, 6, 8, 9, 10] represented as nodes in a TIN, to map a TIN model to a Time Sliced Bayesian Network for incorporating feedback evidence, to determine best course of actions for both timed and un-timed versions of Influence Nets [7, 12] and to assess temporal aspects of the influences on objective nodes [11, 35].

The existing developments of INs and TINs suffer from a number of deficiencies: they do not represent scenarios encompassing dependent conditioning events and they utilize a priori probabilities inconsistently, in violation of the Bayes Rule and the Theory of Total Probability. In this paper, we present a comprehensive theory of Influence Networks, which is free of restrictive independence assumptions, which is consistently observing the Bayes Rule and the Theorem of Total Probability and which also encompasses temporal issues. The organization of the paper is as follows: In Section II, the motivation of the problem is stated. In section III, we present the theoretical formalization and derive initial relationships. In section IV, we derive the dynamic programming evolution of the influence constants. In section V, we examine the case where in the generic model, the affecting events are mutually independent, where in section VI, the case where the latter events form a Markov chain is examined. In section VII, temporal considerations are presented. In section VIII we discuss decision model selection and testing. In section IX, special forms of the influence constants are discussed. In Section X, we discuss evaluation metrics. In section XI, the experimental setup is laid out and some numerical results are presented. In section XII, conclusions are drawn.

II. Problem Motivation

We are concerned with the evaluation of cause-effect relationships between interconnected events. In particular, if the status of some event B is affected by the status of a set of events, A_1 to A_n, we are interested in a qualification and quantification of this effect. We first graph the relationships between events B and A_1 to A_n in a network format, as in Fig. 1 below, with each event being a node, with arcs indicating relationships and with arrows representing the cause-effect directions. This graphical representation is identical to that used in BNs.

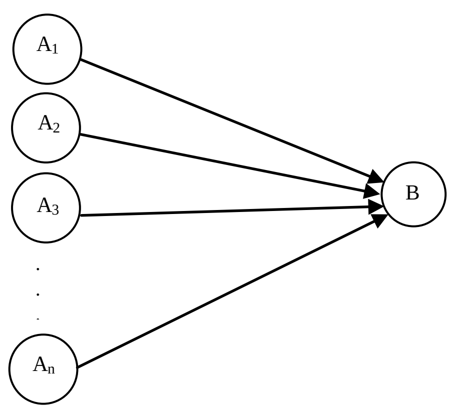


Fig. 1. Cause-Effect Relationships

Given the graph of Fig. 1, we next decide the metric to be used for the quantification of the effects of events A_1 to A_n on event B. As in BNs, modeling each of the involved events as random variables, we use conditional probabilities as effect metrics: in particular, we use the probabilities that event B occurs, given each of the 2^n scenarios regarding the occurrence or nonoccurrence of each one of the events A_1 to A_n.

Upon the decision to use conditional probabilities as the effect metrics, the issue of their computation arises. In most realistic scenarios, there exist insufficient amount of data for the reliable estimation of these probabilities. Instead, some influence indicators may be provided by experts. In the example of Fig. 1, for instance, for each one of the 2^n scenarios regarding the occurrence or nonoccurrence of each on of the events A_1 to A_n, an expert may provide a number between –1 and 1, to reflect his assessment as to the effect of the above scenario on the occurrence of event B. The latter number is named **influence constant**. The objective at this point is to utilize the so provided influence constants for the approximate evaluation and computation of the required conditional probabilities, in a mathematically correct and consistent fashion. These conditional probabilities are subsequently utilized for the probabilistic evaluation of event occurrences in a network of events, giving rise to an Influence Network (IN). In different terms, a IN is a BN whose conditional probabilities are computed via the use of influence constants. The term IN should not be confused with a similarly named formalism called Influence Diagrams [13-15, 26]. Unlike INs, an Influence Diagram (ID) has different types of nodes (i.e., decision nodes, chance nodes, and utility nodes) and different types of influences (i.e., arcs between the nodes); and the decisions in an ID are assumed to have a certain precedence relationship among them. The IDs can be considered a BN extended with a utility function, while a IN, as noted above, is a special instance of a BN whose conditional probabilities are computed via the use of influence constants and which uses a set of special purpose algorithms for calculating the impact of a set of external affecting events on some desired effect/objective node.

Frequently, in several realistic scenarios, assessments of event occurrences may be needed at times when the status of all affecting events may not be known, while such assessments require sequential adaptation, as the status of more affecting events are revealed. For example, in Fig. 1, the evaluation of the probability of event B may be needed at times when the status of only some of the events A are known, while this probability need to be subsequently adapted when the status of the remaining A events become known. Such sequential adaptations require pertinent sequential computation methodologies for the approximation of conditional probabilities via influence constants and give rise to Time Influence Networks (TINs).

Below, we present a theory of INs and TINs that is comprehensive and mathematically consistent and includes the computational methodologies required, as stated above.

III. Initial Modeling and Relationships

In this section, we formalize our approach for the development of INs and TINs.

Let us consider an event B being potentially affected by events $\{A_i\}_{1 \le i \le n}$. In particular, we are interested in the effect the presence or absence of any of the events in the set $\{A_i\}_{1 \le i \le n}$ may have on the occurrence of event B.

Let us first define:

X_1^n: An n-dimensional binary random vector whose j^{th} component is denoted X_j, where $X_j = 1$; if the event A_j is present, and $X_j = 0$; if the event A_j is absent. We will denote by x_1^n realizations or values of the random vector X_1^n.

A given realization x_1^n of the binary vector X_1^n describes precisely the status of the set $\{A_i\}_{1 \le i \le n}$ of events, regarding which events in the set are present. We name the vector X_1^n, the **status vector** of the affecting events. To quantify the effects of the status vector X_1^n on the event B, we define the **influence constant** $h_n(x_1^n)$ via the following quantitative properties

$$h_n(x_1^n) = \begin{cases} 1 & ;\text{if } \begin{array}{l}\text{given n affecting events, given the status}\\ \text{vector } x_1^n\text{, event B occurs surely}\end{array} \\ -1 & ;\text{if } \begin{array}{l}\text{given n affecting events, given the status}\\ \text{vector } x_1^n\text{, the nonoccurrence of event B is sure}\end{array} \\ 0 & ;\text{if } \begin{array}{l}\text{given n affecting events, given the status}\\ \text{vector } x_1^n\text{, the occurrence of event B is unaffected}\end{array} \end{cases} \quad (1)$$

Let $P(B \mid x_1^n)$ denote the probability of occurrence of event B, given the status vector x_1^n. Then, the quantitative definition of

the influence constant $h_n(x_1^n)$ in (1) can be rewritten as follows, where $P(B)$ denotes the unconditional probability of occurrence of the event B.

$$P(B \mid x_1^n) = \begin{cases} 1 \;\; ; \text{ if } & h_n(x_1^n) = 1 \\ P(B); \text{ if } & h_n(x_1^n) = 0 \\ 0 \;\; ; \text{ if } & h_n(x_1^n) = -1 \end{cases} \tag{2}$$

We now extend the definition of all values in $[1,-1]$ of the influence constant, via linear interpolation from (2). In particular, we define the influence constant via its use to determine the derivation of the conditional probability $P(B \mid x_1^n)$ from the unconditional probabilities $P(B)$, where this derivation is derived via linear interpolation from (2). We thus obtain.

$$P(B \mid x_1^n) = \begin{cases} P(B) + h_n(x_1^n)[1 - P(B)]; \text{if } \; h_n(x_1^n) \in [0,1] \\ P(B) + h_n(x_1^n)P(B) \quad ; \text{if } h_n(x_1^n) \in [-1,0] \end{cases} \tag{3}$$

Defining $\operatorname{sgn} \gamma = \begin{cases} 1; \text{if } \; \gamma \geq 0 \\ 0; \text{if } \; \gamma < 0 \end{cases}$, we can finally write (3) as follows

$$P(B \mid x_1^n) = P(B)\{1 + h_n(x_1^n)[1 - P(B)]P^{-1}(B)\}^{\operatorname{sgn} h_n(x_1^n)} \times \{1 + h_n(x_1^n)\}^{1 - \operatorname{sgn} h_n(x_1^n)} \tag{4}$$

At this point, we present a formal definition of INs and TINs.

Definition 1

An Influence Network (IN) is a Bayesian Network mapping conditional probabilities $P(B \mid x_1^n)$ via the utilization of influence constants as in (4)**.** Formally, an Influence Net is a tuple (**V**, **E**, **C**, **A**, **B**), with **G** = (**V**, **E**) representing a directed-acyclic graph satisfying the Markov condition (as in BN), where

V: set of nodes representing binary random variables,

E: set of edges representing causal influences between nodes,

C: set of causal strengths: $E \rightarrow \left\{ \left[h_1^{(i)}(x_i = 1), h_1^{(i)}(x_i = 0) \right] \text{such that } h_1\text{'s} \in [-1,1] \right\}$,

A: a subset of **V** representing *external* affecting events $\{A_i\}_{1 \leq i \leq n}$ and a status of the corresponding vector X_1^n,

B: Probability distribution of the status vector X_1^n corresponding to the external affecting events $\{A_i\}_{1 \leq i \leq n}$.

A Timed Influence Network (TIN) adds two temporal parameters to the definition of a IN. Formally, a TIN is a tuple (**V**, **E**, **C**, **D**, **A$_T$**, **B**), where **V**, **E**, **C**, and **B** are as defined for INs;

D: set of temporal delays on edges: **E** $\rightarrow$ N,

A$_T$: same as **A** with the addition that the status of each external affecting event is *time tagged* representing the time of realization of its status. In the IN/TIN literature [12, 27, 28, 29, 30, 33, 34], A_T is also referred to as a Course of Action (COA). A COA is, therefore, a time-sequenced collection of external affecting events and their status.

Returning to the influence constant notion, we note that there exist 2^n distinct values of the status vector x_1^n; thus, there exist 2^n distinct values of the influence constant $h_n(x_1^n)$ as well as of the conditional probabilities in (4). In the case where the cardinality of the set $\{A_i\}_{1 \leq i \leq n}$ is one, the influence constant $h_1(x_1)$ equals the constant h in [22]; if $x_1 = 1$ and equals the

constant g in [22]; if $x_1 = 0$.

We now proceed with a definition which will lead to a mathematically correct relationship between influence constants and unconditional probabilities.

Definition 2

A IN or TIN model is **consistent** if it observes the Bayes Rule.

Let $P(x_1^n)$ denote the probability of the status vector X_1^n at the value x_1^n. We can then express the following simple lemma.

Lemma 1

Let the influence constant $h_n(x_1^n)$ be accepted as reflecting accurately the relationship between the affecting events $\{A_i\}_{1 \le i \le n}$ and event B. Then the IN or TIN model is consistent iff:

$$\sum_{x_1^n} P\left(x_1^n\right)\left\{1 + h_n(x_1^n)\left[1 - P\left(B\right)\right]P^{-1}\left(B\right)\right\}^{\operatorname{sgn} h_n(x_1^n)}\left\{1 + h_n(x_1^n)\right\}^{1-\operatorname{sgn} h_n(x_1^n)} = 1 \tag{5}$$

Or

$$\sum_{x_1^n} P\left(x_1^n\right) h_n(x_1^n)\left\{\left[1 - P(B)\right]P^{-1}\left(B\right)\right\}^{\operatorname{sgn} h_n(x_1^n)} = 0$$

Proof:

Via the Bayes' Rule, $P\left(B\right) = \sum_{x_1^n} P(x_1^n)P(B \mid x_1^n)$. Substituting expression (4) in the latter expression, we obtain (5).

Expression (5) relates the influence constant $h_n(x_1^n)$ to the unconditional probabilities of event B and the status vector X_1^n. This relationship is necessary if the influence constant is accepted as accurately representing the conditional probability $P(B \mid x_1^n)$ in (3). Generally, the influence constant is selected based on a system design assessment provided by experts, while the a priori probabilities $P(x_1^n)$ are accepted to accurately represent the actual model.

IV. Evolution of the influence Constant

In Section III, we derived the relationship between the conditional probability of event B, and the status x_1^n of its affecting events $\{A_i\}_{1 \le i \le n}$, via the influence constant $h_n(x_1^n)$. This relationship is based on the assumption that $\{A_i\}_{1 \le i \le n}$ is the maximum set of events affecting event B and that the value x_1^n of the status vector is given. In this section we investigate the case where the status of some of the affecting events may be unknown. Towards this direction, we derive a dynamic programming relationship between the influence constants $h_n(x_1^n)$ and $h_{n-1}(x_1^{n-1})$, where $h_{n-1}(x_1^{n-1})$ is the constant corresponding to the case where the status of the affecting event A_n is unknown. We express a lemma whose proof is in the Appendix. The proof is based on the observation of the Bayes Rule and the Theorem of Total Probability.

Lemma 2

Let the probability $P(B)$ be as in Section III and let $P(x_n \mid x_1^{n-1})$ denote the probability of the value of the last bit in the status

vector X_1^n being x_n, given that the reduced status vector value is x_1^{n-1}. Then, the influence constant $h_{n-1}(x_1^{n-1})$ is given as a function of the influence constant $h_n(x_1^n)$, as shown below.

$$h_{n-1}(x_1^{n-1}) = \begin{cases} Q_n & ; \quad Q_n \in [-1,0] \\ P(B)[1-P(B)]^{-1}Q_n & ; \quad Q_n \in [0, P^{-1}(B)-1] \end{cases} \tag{6}$$

where

$$Q_n \overset{\Delta}{=} \sum_{x_n=0,1} P(x_n \mid x_1^{n-1}) \{h_n(x_1^n)\} \{[1-P(B)]P^{-1}(B)\}^{\operatorname{sgn} h_x(x_1^n)} \tag{7}$$

We note that the influence constants are deduced from the same constants of higher dimensionality, as shown in Lemma 2. In accordance, conditional probabilities of the event B are produced from the deduced influence constants, via expression (4), as:

$$P(B \mid x_1^{n-1}) = P(B)\{1 + h_{n-1}(x_1^{n-1})[1-P(B)]P^{-1}(B)\}^{\operatorname{sgn} h_{n-1}(x_1^{n-1})} \times \{1 + h_{n-1}(x_1^{n-1})\}^{1-\operatorname{sgn} h_{n-1}(x_1^{n-1})} \tag{8}$$

It is important to note that in the dynamic programming evolution of the influence constants $h_n(x_1^n)$, as well as in the evolution of the conditional probabilities in (7), knowledge of the joint probability $P(x_1^n)$ is assumed. This reflects a conjecture by the system designer, based on his /her previous experience regarding the a priori occurrence of the affecting events $\{A_i\}_{1 \le i \le n}$. Thus the probability $P(x_1^n)$ used for the construction exhibited by Lemma 2 is a design probability and it may not coincide with the actual probabilities of the status vector X_1^n. When full scale dependence of the components of the status vector X_1^n is incorporated within the design probability $P(x_1^n)$, then the relationship between the different dimensionality influence constants is that reflected by Lemma 2 and is of dynamic programming nature. In the case where the design probability $P(x_1^n)$ generically reflects either a Markov Chain of events or mutually independent events, then the relationships between the different dimensionality influence constants may be also of recursive nature. The cases of Markovian or independent affecting events, as modeled by the system designer, are examined in Sections V and VI below.

V. The case of Independent Affecting Events

In this section, we consider the special case where the affecting events $\{A_i\}_{1 \le i \le n}$ are assumed to be generically mutually independent. Then, the components of the status vector X_1^n are mutually independent, and:

$$P(x_1^n) = \prod_{i=1}^{n} P(x_i) \quad ; \quad P(x_1^n \mid B) = \prod_{i=1}^{n} P(x_i \mid B) \tag{9}$$

Let us denote by $h_1^{(i)}(x_i)$ the influence constant corresponding to the effect of the event A_i on the occurrence of the event B, when event A_i acts in isolation and when the status value of the event is x_i. Then, from expression (4) in Section III, we have:

$$P(B \mid x_i) = P(B)\{1 + h_1^{(i)}(x_i)[1-P(B)]P^{-1}(B)\}^{\operatorname{sgn} h_1^{(i)}(x_i)} \bullet \{1 + h_1^{(i)}(x_i)\}^{1-\operatorname{sgn} h_1^{(i)}(x_i)} \tag{10}$$

We now express a lemma whose proof is in the Appendix.

Lemma 3

Let the events $\{A_i\}_{1 \le i \le n}$ that affect event B be assumed to be generically mutually independent. Then

$$P\left(B \mid x_1^n\right) = P(B)\prod_{i=1}^{n}\left\{1 + h_1^{(i)}(x_i)[1 - P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_1^{(i)}(x_i)} \bullet \left\{1 + h_1^{(i)}(x_i)\right\}^{1-\operatorname{sgn} h_1^{(i)}(x_i)} \tag{11}$$

Via the same logic as that in the last part in the proof of Lemma 1, we can show the result expressed in the corollary below.

Corollary 1

When the affecting events are assumed to be generically mutually independent then, the influence constant $h_n(x_1^n)$ is given as a function of the single event influence constants $\{h_1^{(i)}(x_i)\}_{1 \le i \le n}$, as follows:

$$h_n(x_1^n) = \begin{cases} R_n - 1 & ; \quad \text{if } R_n \in [0,1] \\ P(B)[1 - P(B)]^{-1}[R_n - 1]; & \text{if } R_n \in \left[1, P^{-1}(B)\right] \end{cases} \tag{12}$$

where

$$R_n \stackrel{\Delta}{=} \prod_{i=1}^{n}\left\{1 + h_1^{(i)}(x_i)[1 - P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_1^{(i)}(x_i)} \times \left\{1 + h_1^{(i)}(x_i)\right\}^{1-\operatorname{sgn} h_1^{(i)}(x_i)} \tag{13}$$

The sequence of expressions $\{R_i\}_{1 \le i \le n}$ in (13) is clearly recursively generated and the conditional probability $P(B \mid x_1^n)$ is given by $h_n(x_1^n)$ as in (4) in Section III.

We note that the consistency condition in Lemma 1, Section III reduces in a straight forward fashion and by construction to the following condition here:

$$\sum_{x_i = 0,1} P(x_i)\left\{1 + h_1(x_i)[1 - P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_1(x_i)}\left[1 + h_1(x_i)\right]^{1-\operatorname{sgn} h_1(x_i)} = 1; \forall i$$

Or

$$\sum_{x_i = 0,1} P(x_i) h_1(x_i)\left\{[1 - P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_1(x_i)} = 0; \forall i$$

VI. The case of A Markov Chain of Affecting Events

In this section, we consider the case where the affecting events $\{A_i\}_{1 \le i \le n}$ are assumed to form generically a Markov Chain. In particular, we assume that the design probabilities $P(x_1^n \mid B)$ and $P(x_1^n)$ are such that:

$$\begin{aligned} P(x_1^n \mid B) &= \prod_{i=1}^{n} P(x_i \mid x_{i-1}, B) \\ P(x_1^n) &= \prod_{i=1}^{n} P(x_i \mid x_{i-1}) \end{aligned} \tag{14}$$

where $P(x_1 \mid x_0, B) \stackrel{\Delta}{=} P(x_1 \mid B)$ and $P(x_1 \mid x_0) \stackrel{\Delta}{=} P(x_1)$

We denote by $h_1^{(1)}(x_1)$ the influence constant corresponding to the effect of the event A_1 on the occurrence of the event B, when the status value of A_1, is given by x_1. We denote by $h_2^{(i,i+1)}(x_i, x_{i+1})$ the influence constant corresponding to the effect of the events A_i and A_{i+1} on the occurrence of the event B, when the status values of the (A_i, A_{i+1}) pair are given by (x_i, x_{i+1}). Then, via (4) in Section III, we have

$$P(B \mid x_1) = P(B)\left\{1 + h_1^{(1)}(x_1)[1-P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_1^{(1)}(x_1)} \times \left\{1 + h_1^{(1)}(x_1)\right\}^{1-\operatorname{sgn} h_1^{(1)}(x_1)} \tag{15}$$

$$\begin{aligned} P(B \mid x_i, x_{i+1}) = P(B)\left\{1 + h_2^{(i,i+1)}(x_i, x_{i+1})[1-P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_2^{(i,i+1)}(x_i,x_{i+1})} \times \\ \left\{1 + h_2^{(i,i+1)}(x_i, x_{i+1})\right\}^{1-\operatorname{sgn} h_2^{(i,i+1)}(x_i,x_{i+1})}; i = 1 \end{aligned} \tag{16}$$

We now express a lemma whose proof is in the Appendix.

Lemma 4

Let the affecting events $\{A_i\}_{1 \le i \le n}$ be assumed to generically form a Markov Chain; thus, $P(x_1^n)$ is assumed to satisfy the equation in (13). Then,

$$\begin{aligned} P(B \mid x_1^n) &= P(B)\left\{1 + h_1^{(1)}(x_1)[1-P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_1^{(1)}(x_1)} \times \left\{1 + h_1^{(1)}(x_1)\right\}^{1-\operatorname{sgn} h_1^{(1)}(x_1)} \times \\ &\prod_{i=2}^{n} \frac{\left\{1 + h_2^{(i,i-1)}(x_i, x_{i-1})[1-P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_2^{(i,i-1)}(x_i,x_{i-1})} \times \left\{1 + h_2^{(i,i-1)}(x_i, x_{i-1})\right\}^{1-\operatorname{sgn} h_2^{(i,i-1)}(x_i,x_{i-1})}}{\left\{1 + h_1^{(i-1)}(x_{i-1})[1-P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_1^{(i-1)}(x_{i-1})} \times \left\{1 + h_1^{(i-1)}(x_{i-1})\right\}^{1-\operatorname{sgn} h_1^{(i-1)}(x_{i-1})}} \\ &\stackrel{\Delta}{=} P(B)W_n \end{aligned} \tag{17}$$

where,

$$h_1^{(i)}(x_i) = \begin{cases} Q_{i,i+1} - 1 & ; \quad \text{if } Q_{i,i+1} \in [0,1] \\ P(B)[1-P(B)]^{-1}[Q_{i,i+1} - 1] & ; \quad \text{if } Q_{i,i+1} \in [1, P^{-1}(B)] \end{cases} \tag{18}$$

$$Q_{i,i+1} \stackrel{\Delta}{=} \sum_{x_{i+1}=0,1} P(x_{i+1} \mid x_i)\left\{1 + h_2^{(i,i+1)}(x_i, x_{i+1})[1-P(B)]P^{-1}(B)\right\}^{\operatorname{sgn} h_2^{(i,i+1)}(x_i,x_{i+1})} \times \left\{1 + h_2^{(i,i+1)}(x_i, x_{i+1})[1+P(B)]P^{-1}(B)\right\}^{1-\operatorname{sgn} h_2^{(i,i+1)}(x_i,x_{i+1})} \tag{19}$$

As with Corollary 1 in Section V, we can express the corollary below, in a direct fashion.

Corollary 2

When the affecting events $\{A_i\}_{1 \le i \le n}$ are assumed to generically form a Markov Chain, depicted by the expression in (13), then, the influence constant $h_n(x_1^n)$ is given as a function of the influence constants $\{h_1^{(i)}(x_i)\}$ and $\{h_2^{(i,i-1)}(x_i, x_{i-1})\}$, as below, where W_n is defined in (17)

$$h_n(x_1^n) = \begin{cases} W_n - 1 & ; \quad \text{if } W_n \in [0,1] \\ P(B)[1-P(B)]^{-1}[W_n - 1]; & \text{if } W_n \in [1, P^{-1}(B)] \end{cases} \tag{20}$$

The sequence $\{W_i\}_{1 \le i \le n}$ in (17) is clearly recursively expressed; thus, $h_n(x_1^n)$ is recursively evolving. The consistency condition in Lemma 1, Section III, takes here the following form, by construction.

$$\sum_{x_i=0,1}\sum_{x_{i-1}=0,1}P(x_i \mid x_{i-1})\left\{1+h_2^{(i,i-1)}(x_i x_{i-1})[1-P(B)]P^{-1}(B)\right\}^{\mathrm{sgn}\, h_2^{(i,i-1)}(x_i,x_{i-1})} \times \left\{1+h_2^{(i,i-1)}(x_i,x_{i-1})\right\}^{1-\mathrm{sgn}\, h_2^{(i,i-1)}(x_i,x_{i-1})} = 1; \forall i$$

VII. Temporal Extension

In Sections III and IV, we presented our theoretical foundation for the development of INs and TINs, while in Sections V and VI, we focused on the special cases of independent and Markovian affecting events. In this section, we focus on the formalization of the temporal issues involved in the development of TINs. In particular, we are investigating the dynamics of the relationship of the affecting events $\{A_i\}_{1 \le i \le n}$ to the affected event B, when the status of the former evens are learned asynchronously in time. Without lack in generality – to avoid cumbersome notation – let the affecting events $\{A_i\}_{1 \le i \le n}$ be ordered in the order representing the time when their status become known. That is, the status of events A_1 is first known, then that of event A_2, and so on. In general, the status of event A_k becomes known after the status of the events $A_1 \ldots, A_{k-1}$ are known, and this knowledge becomes available one event at the time.

Let us assume that the considered system model implies full dependence of the components of the status vector X_1^n. Then, the influence constants $\{h_i(x_1^n)\}_{1 \le i \le n-1}$ are first pre-computed via the dynamic programming expression in Lemma 2, Section IV, utilizing the pre-selected a priori probabilities $P(x_1^n)$ that are part of the given system parameters. The above influence constants can be recursively computed if the adopted system model implies either generically independent affecting events or affecting events that generically form a Markov Chain, as shown in Sections V and VI.

Let T_0 denote the time when the computation of the system dynamics starts. Let T_1 denote the time when the status of event A_1, becomes known. Let T_k; $1 \le k \le n$ denote the time when the status of event A_k becomes known. Then at time T_k, the conditional probabilities $P(B \mid x_1^k)$ are computed via expression (4), Section III, as,

$$\underset{i\le k\le n}{P}\left(B \mid x_1^k\right) = P(B)\left\{1+h_k(x_1^k)[1-P(B)]P^{-1}(B)\right\}^{\mathrm{sgn}\, h_k(x_1^k)} \times \left\{1+h_k(x_1^k)\right\}^{1-\mathrm{sgn}\, h_k(x_1^k)} \qquad (21)$$

; where the probability $P(B)$ is computed via the consistency condition (5).

As the knowledge about the status of the affecting events unravels, the conditional probabilities of event B in (21) evolve dynamically in time and finally converge to the probability $P(B \mid x_1^n)$ at time T_n, when the status of all the affecting events become known.

It is important to point out that the conditional probability in (21) is sensitive to the time ordering of the affecting events. That is, for the same value x_1^k of a partial affecting vector, but different time ordering of events, different conditional probabilities values of the affected event B arise. Thus, the order by which the status of the affecting events become known is crucial in the evaluation of the conditional probabilities of event B.

VIII. Selection and Testing of the Decision Model

A. *Model Selection*

As we have discussed earlier, the unconditional probabilities $P(x_1^n)$ as well as the influence constant $h_n(x_1^n)$ are design parameters that may not represent the actual parameters correctly. Furthermore, as discussed in Section III, the design parameters must be **consistent**, where consistency is represented by the satisfaction of condition (5) in Lemma 1. Condition (5) can be

rewritten as follows, in a straightforward fashion.

$$[1-P(B)]\sum_{x_1^n:\mathrm{sgn}h_n(x_1^n)=1} P\left(x_1^n\right)h_n(x_1^n) = P(B)\sum_{x_1^n:\mathrm{sgn}h_n(x_1^n)=0} P\left(x_1^n\right)\left|h_n(x_1^n)\right| \tag{22}$$

which gives:

$$P(B)=\left[\sum_{x_1^n:\mathrm{sgn}h_n(x_1^n)=1} P\left(x_1^n\right)h_n(x_1^n)\right]\left[\sum_{x_1^n} P\left(x_1^n\right)\left|h_n(x_1^n)\right|\right]^{-1};\ \text{when } \sum_{x_1^n} P\left(x_1^n\right)\left|h_n(x_1^n)\right| \neq 0 \tag{23}$$

Example:

Let us consider the case where the only affecting event for B is A_1. Let $P(A_1)\stackrel{\Delta}{=}P(X_1=1)=p$, where then, $P(A_1^C)\stackrel{\Delta}{=}P(X_1=0)=1-p$. Define h and g as in [22] and let $P(B)$ be what has been called in [22] base probability for the event B. Then, due to (22) the above parameters must satisfy the following equation(s):

$$\begin{array}{lll} either & [1-P(B)]ph = P(B)(1-p)|g| & ;\ \text{if } h>0 \text{ and } g<0 \\ or & [1-P(B)](1-p)g = P(B)p\,|h| & ;\ \text{if } h<0 \text{ and } g>0 \end{array}$$

no other h and g combinations are acceptable.

When new information about the a priori probability $P(x_1^n)$ is obtained, then, $P(B)$ and/or $h_n(x_1^n)$ need to be accordingly adjusted to satisfy the condition in (22). We note that the latter condition involves a number of free parameters; thus even specification of the probabilities $P(B)$ and $P(x_1^n)$ does not specify uniquely the values of the influence constant $h_n(x_1^n)$. Naturally, specification of $P(x_1^n)$ and $h_n(x_1^n)$ uniquely determines the probability $P(B)$, however, as in (23).

In the case that the assumed system design model implies generically independent affecting events $\{A_i\}_{1\le i\le n}$, then, for consistency the probability $P(B)$, the probability $P(x_1^n)=\prod_{i=1}^{n}P(x_i)$ of the status vector and the influence constants $\{h_1^{(i)}(x_i)\}$ are constraint to satisfy the condition:

$$\sum_{x_i=0,1} P\left(x_i\right)\left\{1+h_1(x_i)[1-P(B)]P^{-1}(B)\right\}^{\mathrm{sgn}h_1(x_i)}\left\{1+h_1(x_i)\right\}^{1-\mathrm{sgn}h_1(x_i)}=1;\forall i \tag{24}$$

Or

$$\sum_{x_i=0,1} P\left(x_i\right)h_1(x_i)\left\{[1-P(B)]P^{-1}(B)\right\}^{\mathrm{sgn}h_1(x_i)}=0;\forall i$$

B. Model Testing

Since the "consistency" constraints allow for a number of free parameters, we will focus on the influence constant $h_n(x_1^n)$ as the constant to be tested, when information about the probabilities of the events $\{A_i\}_{1\le i\le n}$ and B is obtained. Thus, model testing will involve comparison of the $P(x_1^n)$ and $P(B)$ probabilities assumed in the model with those computed, to test the validity of

the assumed influence constant. When the computed $P(x_1^n)$ and $P(B)$ values do not satisfy equation (22) for the assumed $h_n(x_1^n)$, then a non valid model is declared and a new influence constant $h_n(x_1^n)$ is sought, in satisfaction of the consistency condition in (22).

IX. Some Special Influence Constants

As noted at the end of Section VIII, the influence constant is a important component of the system model: the appropriate choice of this constant needs to be carefully thought out, to accurately reflect the interleaving of partial influences. In this section, we study some specific influence constants, $h_n(x_1^n)$. In particular, we study such constants that are specific analytic functions of the one-dimensional components $h_i(x_i)$; $1 \le i \le n$. We note that we are not mapping the $\{h_i(x_i)\}_{1\le i\le n}$ constants onto conditional probabilities $\{P(B \mid x_i)\}_{1\le i\le n}$. Instead, we are using the constants $\{h_i(x_i)\}_{1\le i\le n}$ to construct a global $h_n(x_1^n)$ influence constant; it is the latter constant which is mapped onto the conditional probability $P(B \mid x_1^n)$, as in Section III.

A. *The $h_n(x_1^n)$ corresponding to the CAST logic*

The influence constant presented below is that used by the CAST logic in [2, 22-25].

In the present case, given the constants $\{h_i^{(i)}(x_i)\}_{1\le i\le n}$ the global influence constant, $h_n(x_1^n)$, is defined as follows

$$h_n(x_1^n) = \left[\prod_{i:h_1(x_i)<0}\left(1-\left|h_1^{(i)}(x_i)\right|\right) - \prod_{i:h_1(x_i)>0}\left(1-\left|h_1^{(i)}(x_i)\right|\right)\right] \times \left[\max\left(\prod_{i:h_1(x_i)<0}\left(1-\left|h_1^{(i)}(x_i)\right|\right), \prod_{i:h_1(x_i)>0}\left(1-\left|h_1^{(i)}(x_i)\right|\right)\right)\right]^{-1} \quad (25)$$

In agreement with the results in Section III, and via (5) in Lemma 1, the global constants $h_n(x_1^n)$ and the probabilities $P(x_1^n)$ and $P(B)$ must satisfy the consistency condition

$$\sum_{x_1^n} P\left(x_1^n\right)\left\{1 + h_n(x_1^n)\left[1-P(B)\right]P^{-1}(B)\right\}^{\operatorname{sgn} h_n(x_1^n)} \left\{1 + h_n(x_1^n)\right\}^{1-\operatorname{sgn} h_n(x_1^n)} = 1 \quad (26)$$

Via (4), the conditional probabilities $P(B \mid x_1^n)$ are then given, by the following expression:

$$P\left(B \mid x_1^n\right) = P(B)\left\{1 + h_n(x_1^n)\left[1-P(B)\right]P^{-1}(B)\right\}^{\operatorname{sgn} h_n(x_1^n)} \times \left\{1 + h_n(x_1^n)\right\}^{1-\operatorname{sgn} h_n(x_1^n)} \quad (27)$$

For maintaining the consistency condition in (26), the conditional probability $P(B \mid x_1^{n-1})$ is defined via the influence constant $h_{n-1}(x_1^{n-1})$ as in Lemma 2, Section III, where,

$$P(B \mid x_1^{n-1}) = P(B)\left\{1 + h_{n-1}(x_1^{n-1})\left[1-P(B)\right]P^{-1}(B)\right\}^{\operatorname{sgn} h_{n-1}(x_1^{n-1})} \times \left\{1 + h_{n-1}(x_1^{n-1})\right\}^{1-\operatorname{sgn} h_{n-1}(x_1^{n-1})}$$

and

$$h_{n-1}(x_1^{n-1}) = \begin{cases} Q_n - 1 & ; \quad Q_n \in [0,1] \\ P(B)\left[1-P(B)\right]^{-1}\left[Q_n - 1\right] & ; \quad Q_n \in \left[1, P^{-1}(B)\right] \end{cases}$$

$$Q_n \stackrel{\Delta}{=} \sum_{x_n=0,1} P(x_n \mid x_1^{n-1})\left\{1+h_n(x_1^n)\left[1-P(B)\right]P^{-1}(B)\right\}^{\operatorname{sgn} h_n(x_1^n)} \times \left[1+h_n(x_1^n)\right]^{1-\operatorname{sgn} h_n(x_1^n)}$$

B. A $h_n(x_1^n)$ *Constant Representing Extreme Partial Values*

In this part, we first define the effect of the constants $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$ on the event B as follows:

a. If at least one of the constants $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$ equals the value 1, then event B occurs surely, if in addition $\sum_{i=1}^{n} h_1^{(i)}(x_i) > 0$

b. If at least one of the constants $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$ equals the value -1, then the nonoccurrence of event B is sure, if in addition $\sum_{i=1}^{n} h_1^{(i)}(x_i) < 0$

c. The events $\{A_i\}_{1 \le i \le n}$ do not effect the event B if $\sum_{i=1}^{n} h_1^{(i)}(x_i) = 0$

The above conditions translate to the following initial expressions for the conditional probability $P(B \mid x_1^n)$, where x_1^n is the value of the status vector of the affecting events $\{A_i\}_{1 \le i \le n}$:

$$P(B \mid x_1^n) = \begin{cases} 1 & ; if \quad \max_{1\le i\le n} h_1^{(i)}(x_i) = 1 \text{ and } \sum_{\mathrm{i}=1}^{\mathrm{n}} h_1^{(i)}(x_i) > 0 \\ P(B) & ; if \quad \sum_{\mathrm{i}=1}^{\mathrm{n}} h_1^{(i)}(x_i) = 0 \\ 0 & ; if \quad \min_{1\le i\le n} h_1^{(i)}(x_i) = -1 \text{ and } \sum_{\mathrm{i}=1}^{\mathrm{n}} h_1^{(i)}(x_i) < 0 \end{cases} \tag{28}$$

Via linear interpolation from the above expression we obtain the general expression of the conditional probability $P(B \mid x_1^n)$, as a function of the influence constants $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$, as follows:

$$P(B \mid x_1^n) = \begin{cases} P(B) + \max_{1\le i\le n}\left(h_1^{(i)}(x_i)\right)\left[1-P(B)\right] & ; \quad \sum_{\mathrm{i}=1}^{\mathrm{n}} h_1^{(i)}(x_i) > 0 \\ P(B) & ; \quad \sum_{\mathrm{i}=1}^{\mathrm{n}} h_1^{(i)}(x_i) = 0 \\ P(B) + \min_{1\le i\le n}\left(h_1^{(i)}(x_i)\right)P(B) & ; \quad \sum_{\mathrm{i}=1}^{\mathrm{n}} h_1^{(i)}(x_i) < 0 \end{cases} \tag{29}$$

Defining the operators $O(x) \stackrel{\Delta}{=} \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x < 0 \end{cases}$ and $U(x) \stackrel{\Delta}{=} \begin{cases} 1 & ; \quad x \ge 0 \\ 0 & ; \quad x < 0 \end{cases}$, we can rewrite equation (26) in a compressed form as follows.

$$P(B \mid x_1^n) = P(B)\left\{1 + P^{-1}(B)\left[1 - P(B)\right]\max_{1\le i\le n} h_1^{(i)}(x_i)\right\}^{O(\sum_{i=1}^{n} h_1^{(i)}(x_i))} \left\{1 + \min_{1\le i\le n} h_1^{(i)}(x_i)\right\}^{1-U(\sum_{i=1}^{n} h_1^{(i)}(x_i))} \quad (30)$$

Next, we express a lemma regarding the consistency condition for our present model, evolving from the application of the Bayes Rule and the Theorem of Total Probability on (30). The lemma is the parallel to Lemma 1 in Section III, for the model in the present case.

Lemma 5

For the influence model expressed in (30), the probabilities $P(B)$, $P(x_1^n)$ and the influence constants $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$ must satisfy the following condition:

$$[1 - P(B)] \sum_{x_1^n : \sum_{i=1}^{n} h_1^{(i)}(x_i) > 0} P\left(x_1^n\right) \max_{1\le i\le n} h_1^{(i)}(x_i) + P(B) \sum_{x_1^n : \sum_{i=1}^{n} h_1^{(i)}(x_i) < 0} P\left(x_1^n\right) \min_{1\le i\le n} h_1^{(i)}(x_i) = 0 \quad (31)$$

From the consistency condition in (31), we notice that when examining all the values of the status vector X_1^n, it is necessary that some x_1^n vector values exist such that $\max_{1\le i\le n} h_1^{(i)}(x_i)$ is positive and that some x_1^n vector values exists such that $\min_{1\le i\le n} h_1^{(i)}(x_i)$ is negative.

1) Temporal Issues

Here, we will assume that the very existence of the affecting events is revealed sequentially. Let then the existence and the status of the events $\{A_i\}_{1 \le i \le n}$ be revealed sequentially in time, from A_1 to A_n, where the status of events A_1 to A_k is known at time T_k. At time T_k, the partial status vector x_1^k is expressed and for each one of its values, the probability $P(x_1^k)$ and the quantities, $S_k(x_1^k) \stackrel{\Delta}{=} \sum_{i=1}^{k} h_1^{(i)}(x_i)$, $F_k(x_1^k) \stackrel{\Delta}{=} \max_{1\le i\le k} h_1^{(i)}(x_i)$ and $G_k(x_1^k) \stackrel{\Delta}{=} \min_{1\le i\le k} h_1^{(i)}(x_i)$ are computed. Next, the probability $P(B)$ is computed from (31) as follows:

$$P(B) \stackrel{\Delta}{=} P_k(B) = \left[\sum_{x_1^k : S_k(x_1^k) > 0} P(x_1^k) F_k(x_1^k) - \sum_{x_1^k : S_k(x_1^k) < 0} P(x_1^k) G_k(x_1^k)\right]^{-1} \sum_{x_1^k : S_k(x_1^k) > 0} P(x_1^k) F_k(x_1^k) \quad (32)$$

Given each x_1^k value, the probability $P(B)$ in (32) is then used to compute the conditional probability $P(B \mid x_1^k)$, as,

$$P(B \mid x_1^k) = P_k(B)\left\{1 + P_k^{-1}(B)\left[1 - P_k(B)\right] F_k(x_1^k)\right\}^{O(S_k(x_1^k))} \left\{1 + G_k(x_1^k)\right\}^{1-U(S_k(x_1^k))} \quad (33)$$

At time T_{k+1}, upon the revelation of the existence and the status of the affecting event A_{k+1}, for each status vector x_1^{k+1}, the quantities, $S_{k+1}(x_1^{k+1}) = S_{k+1}(x_1^k) + x_{k+1}$, $F_{k+1}(x_1^{k+1}) = \max\left(F_k(x_1^k), h(x_{k+1})\right)$, $G_{k+1}(x_1^{k+1}) = \min\left(G_k(x_1^k), h_1(x_{k+1})\right)$ are first recursively computed. Then, the probability $P(B)$ is recomputed as

$$\begin{aligned} P(B) \stackrel{\Delta}{=} P_{k+1}(B) = &\left[\sum_{x_1^{k+1} : S_{k+1}(x_1^{k+1}) > 0} P(x_1^{k+1}) F_{k+1}(x_1^{k+1}) - \sum_{x_1^{k+1} : S_{k+1}(x_1^{k+1}) < 0} P(x_1^{k+1}) G_{k+1}(x_1^{k+1})\right]^{-1} \times \\ &\sum_{x_1^{k+1} : S_{k+1}(x_1^{k+1}) > 0} P(x_1^{k+1}) F_{k+1}(x_1^{k+1}) \end{aligned} \quad (34)$$

The probability in (31) is used to compute the conditional probability below.

$$P(B \mid x_1^{k+1}) = P_{k+1}(B)\left\{1 + P_{k+1}^{-1}(B)[1 - P_{k+1}(B)]F_{k+1}(x_1^{k+1})\right\}^{O(S_{k+1}(x_1^{k+1}))}\left\{1 + G_{k+1}(x_1^{k+1})\right\}^{1-U(S_{k+1}(x_1^{k+1}))} \quad (35)$$

We note that the time evolution of the conditional probabilities $P(B \mid x_1^k)$ is different for different time orderings of the affecting events $\{A_i\}_{1 \le i \le n}$.

C. A linear $h_n(x_1^n)$ Constant

Here, we assume that the effects of events $\{A_i\}_{1 \le i \le n}$ on event B are weighted by a known set $\{w_i\}_{1 \le i \le n}$ of weights, such that $w_i \ge 0; \forall i$ and $\sum_{i=1}^{n} w_i = 1$. Then, given the constants $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$, we define $h_n(x_1^n)$ as follows, for some given value $\alpha : 0 \le \alpha < 1$:

$$h_n(x_1^n) = \begin{cases} (1-\alpha)^{-1}\sum_{i=1}^{n} w_i h_1^{(i)}(x_i) & ; \quad \left|\sum_{i=1}^{n} w_i h_1^{(i)}(x_i)\right| \le 1-\alpha \\ 1 & ; \quad \sum_{i=1}^{n} w_i h_1^{(i)}(x_i) \ge 1-\alpha \\ -1 & ; \quad \sum_{i=1}^{n} w_i h_1^{(i)}(x_i) \le -(1-\alpha) \end{cases}$$

A nonzero α value translates to the probability of event B being equal to one, not only when all the $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$ values equal one, but also when a predefined weighted majority exceeds a total weighted sum of $1-\alpha$. Similarly then, the event B occurs with zero probability when the weighted sum of the $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$ values is less than $-(1-\alpha)$, rather than only when it equals -1. The relationships between the $h_n(x_1^n)$ and $h_{n-1}(x_1^{n-1})$ influence constants and the probabilities $P(B)$, $P(x_1^n)$ and $P(B \mid x_1^n)$ are as in IX.A.

D. A $h_n(x_1^n)$ constant representing Noisy OR Format

Given the constants $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$, we define here $h_n(x_1^n)$ as follows; where α is such that $0 \le \alpha \le 1$:

$$h_n(x_1^n) = \left\{1 - (1-\alpha)^{-1}\prod_{i=1}^{n}\left(1 - \left|h_1^{(i)}(x_i)\right|\right)\right\}^{\mathrm{sgn}\left(h_n(x_1^n)\right)}\left\{-1 + \alpha^{-1} - \alpha^{-1}\prod_{i=1}^{n}\left(1 - \left|h_1^{(i)}(x_i)\right|\right)\right\}^{1-\mathrm{sgn}\left(h_n(x_1^n)\right)} \quad (36)$$

Then, via (3) and (5) in Section III, we obtain:

$$P(B) = \alpha \quad (37)$$

$$1 - P(B \mid x_1^n) = \prod_{i=1}^{n}\left(1 - \left|h_1^{(i)}(x_i)\right|\right) \quad (38)$$

The expression in (38) represents the Noisy-OR format [1, 4], where the probabilities in the latter are here substituted by the absolute values of the one-dimensional influence constants $\{h_1^{(i)}(x_i)\}_{1\le i\le n}$.

X. Evaluation Metrics

As already repeatedly stated, the INs and TINs studied in this paper are basically BNs whose conditional probabilities are approximated by expert provided influence constants. Thus, the architectural and computational complexities involved are similar to those in BNs [3, 16, 17, 18, 21], while the complexities involved in the computation of influence constants depend on the specific structure of the latter (see Section IX). The evolution of lower dimensionality conditional probabilities from high dimensionalities ones, as in Lemma 2, Section IV, is of dynamic programming nature inducing polynomial complexity. As stated in Section VIII, the accuracy of a IN or TIN model is determined by the accuracy of the selected influence constants. The accuracy of the latter may be tested and they may be subsequently adjusted appropriately.

XI. Experimental Setup and Numerical Evaluations

A. *Experimental Setup*

In this section, we lay out the steps involved in an experimental setup.

a. Given an event B, determine **all** the events $\{A_i\}_{1 \le i \le n}$ **known** to be affecting its occurrence.

b. Given B, all the known affecting events $\{A_i\}_{1 \le i \le n}$, and the causal strengths $\left[h_1^{(i)}(x_i = 1), h_1^{(i)}(x_i = 0)\right]$ between each A_i and B, design an influence constant $h_n(x_1^n)$, where x_1^n signifies the value of the status vector of the events $\{A_i\}_{1 \le i \le n}$, and where $-1 \le h_n(x_1^n) \le 1; \forall\, x_1^n$ values. The $h_n(x_1^n)$ constant may have one of the forms presented in section IX.

c. If **all** in (b) is given, then upon a given probability of the status vector X_1^n, say $P(x_1^n)\,; \forall\, x_1^n$ values, the probability of event B is given by the following equation, named the consistency equation.

$$\sum_{x_1^n} P\left(x_1^n\right)\left\{1 + h_n(x_1^n)\left[1 - P\left(B\right)\right]P^{-1}\left(B\right)\right\}^{\operatorname{sgn} h_n(x_1^n)} \left\{1 + h_n(x_1^n)\right\}^{1-\operatorname{sgn} h_n(x_1^n)} = 1$$

whose equivalent form is:

$$P(B) = \left[\sum_{x_1^n : \operatorname{sgn} h_n(x_1^n) = 1} P\left(x_1^n\right) h_n(x_1^n)\right]\left[\sum_{x_1^n} P\left(x_1^n\right)\left|h_n(x_1^n)\right|\right]^{-1}$$, if the denominator is non zero

d. When **all** the affecting events $\{A_i\}_{1 \le i \le n}$ are known, but the status of some of them are unknown, then, the probability $P(B)$, as computed in step (c) is used to compute the conditional probability $P(B \mid x_1^k)$, when the status vector of only k affecting events is known as:

$$P(B \mid x_1^k) = P(B)\{1 + h_k(x_1^k)[1 - P(B)]P^{-1}(B)\}^{\operatorname{sgn} h_k(x_1^k)} \bullet \{1 + h_k(x_1^k)\}^{1-\operatorname{sgn} h_k(x_1^k)}$$

where $h_k(x_1^k)$ is computed in a dynamic programming fashion from the influence constant $h_n(x_1^n)$ in (b); as follows:

$$h_{n-1}(x_1^{n-1}) = \begin{cases} Q_n - 1 & ; \quad Q_n \in [0,1] \\ P(B)[1 - P(B)]^{-1}[Q_n - 1] & ; \quad Q_n \in [1, P^{-1}(B)] \end{cases}$$

for $$Q_n \stackrel{\Delta}{=} \sum_{x_n = 0,1} P(x_n \mid x_1^{n-1})\left[1 + h_n(x_1^n)[1 - P(B)]P^{-1}(B)\right]^{\operatorname{sgn} h_n(x_1^n)} \bullet \left[1 + h_n(x_1^n)\right]^{1-\operatorname{sgn} h_n(x_1^n)}$$

We note that in the above expression , the affecting events $\{A_i\}_{1 \le i \le n}$ are assumed ordered as of the revealing of their status in time. Different such ordering results in different evolutions of the conditional probabilities $P(B \mid x_1^k)$.

e. When the existence as well as the status of the affecting events are sequentially revealed, then at time k, $P_k(B)$ and $P_k(B \mid x_1^k)$ are computed as in (c) and (d) where n is substituted by k in the latter.

Example 1:

The following example illustrates the steps (a) to (e) with the help of an example TIN.

i. Fig. 2 shows a IN with a binary event B known to be affected by the events $\{A_i\}_{1 \le i \le 4}$.

ii. The edges connecting the external affecting events $\{A_i\}_{1 \le i \le 4}$ to the event B are shown in Fig. 2, annotated with the constants $\left[h_1^{(i)}(x_i = 1), h_1^{(i)}(x_i = 0)\right]$ for each *i*, where $x_i = 0, 1$ represents one of the two states of an affecting event A_i. A global influence constant $h_4(x_1^4)$ is then designed using the CAST logic expression (25) in Section IX. Table 1 shows the computed values for $h_4(x_1^4)\,;\forall\, x_1^n$.

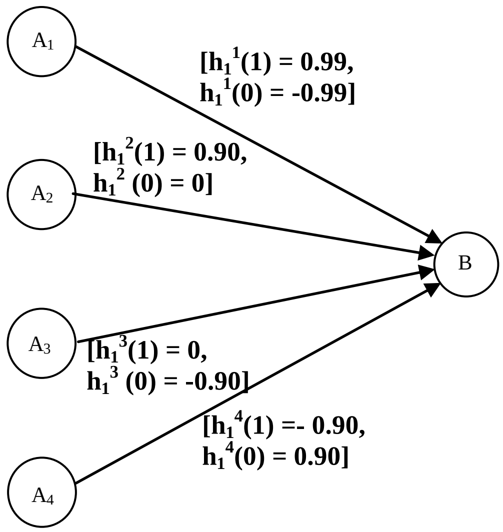


Fig. 2. Example TIN

iii. The joint probability $P(x_1^4)\,;\forall\, x_1^4$ values are computed by assigning $P(x_i = 1) = P(x_i = 0) = 0.5\,;\forall\, i$ and by assuming $\{A_i\}_{1 \le i \le 4}$ to be mutually independent (Lemma 3). The probability of occurrence of event B, i.e., z = 1, is now calculated with the consistency equation, and is given as $P(z = 1) = 0.5$. Assuming the status of all the affecting events to be known, the conditional probabilities $P(B \mid x_1^4);\forall x_1^4$ are calculated via expression (26), and are shown in Table 1.

TABLE 1

x_1	x_2	x_3	x_4	$h_4(x_1^4)$	$P(z = 1 \mid x_1^4)$
0	0	0	0	-0.990000	0.005000
0	0	0	1	-0.999900	0.000050
0	0	1	0	-0.900000	0.050000
0	0	1	1	-0.999000	0.000500
0	1	0	0	-0.900000	0.050000
0	1	0	1	-0.999000	0.000500
0	1	1	0	-0.000001	0.499999
0	1	1	1	-0.990000	0.005000
1	0	0	0	0.990000	0.995000
1	0	0	1	0.000001	0.500001
1	0	1	0	0.999000	0.999500
1	0	1	1	0.900000	0.950000
1	1	0	0	0.999000	0.999500
1	1	0	1	0.900000	0.950000

1	1	1	0	0.999900	0.999950
1	1	1	1	0.990000	**0.995000**

iv. The assumption in step (iii), regarding the knowledge of the status of all the affecting events, may not be valid at times. Such is the case of a TIN with delays on edges, reflecting variations in the times when the status of the affecting events become known (see Definition 1). To illustrate this notion, we add temporal information to the IN in Fig. 2. The added temporal information together with the underlying graph is shown in Fig. 3. The time assigned to an affecting event A_i is the instance at when it assumes a state, i.e., $x_i = 0$ or 1. Prior to that time, the state of the event is assumed unknown. As stated in Definition 1, this combination of the external affecting events' status and their timing is also termed a Course of Action (COA), in the TIN literature.

v. The temporal information in the TIN, Fig. 3, determines the dynamics of the relationship between the affecting events and the affected event B; specifically, the times when the status of the affecting events are revealed to B. Fig. 4 shows a IN equivalent, obtained by mapping the status of the affecting events and their effects on the event B, on a timeline. This mapping determines the number of affecting events 'k' at different time points (or time slices). For the temporal case presented in section VIII, the existence of all the affecting events is known to the event B a priori; their status, however, remain unknown until revealed, as determined by the COA and the delays on the edges. The probability $P(B)$, as calculated in step (c), is used to compute the conditional probabilities $P(B \mid x_1^k); k = 1,2,4$, i.e., $P(B \mid x_1^1)$, $P(B \mid x_1^2)$, and $P(B \mid x_1^4)$,as illustrated in the figure. Table 2 shows the values for $P(B \mid x_1^1)$ and $P(B \mid x_1^2)$, as computed by the corresponding $h_1(x_1^1)$ and $h_2(x_1^2)$. The posterior probability of B captures the impact of an affecting event on B and can be plotted as a function of time for a corresponding COA. This plot is called a Probability Profile [12, 27]. Fig. 5 shows the resulting probability profile for the illustrative example. The plotted values in the profile are shown with bold letters in Tables 1-2. The overall complexity is polynomial.

vi. For the temporal case presented in section IX, the existence as well as the status of the affecting events are not known a priori but are determined by the given COA and the delays on the edges. At time k, $P_k(B)$ and $P_k(B \mid x_1^k)$ are computed as in (c) and (d) where n is substituted by k in the latter. Table 3 shows the computed values of $P_k(B)$ and $P_k(B \mid x_1^k); k = 1,2,4$ and Fig. 5(b) shows the resulting probability profile.

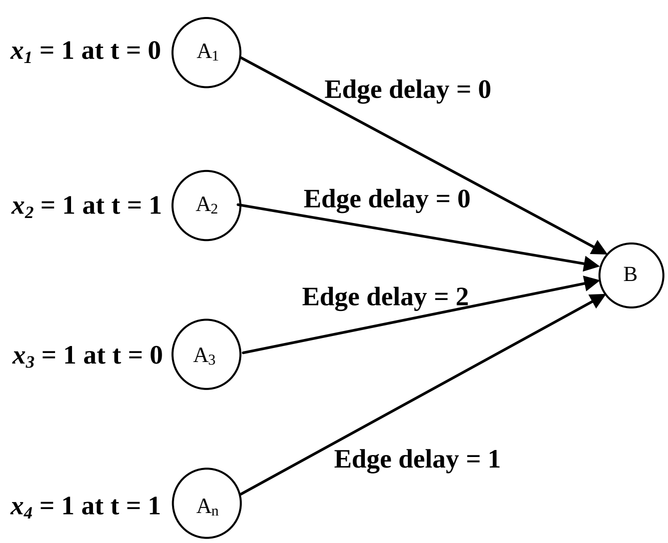


Fig. 3. Example TIN with COA and Edge Delays

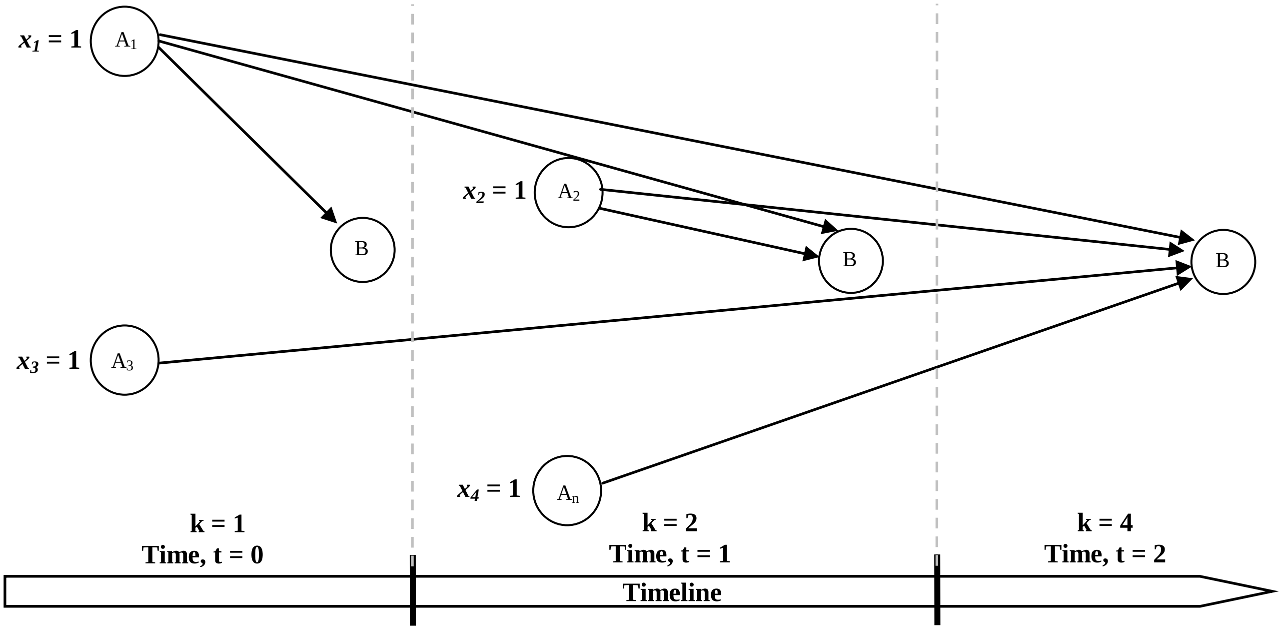


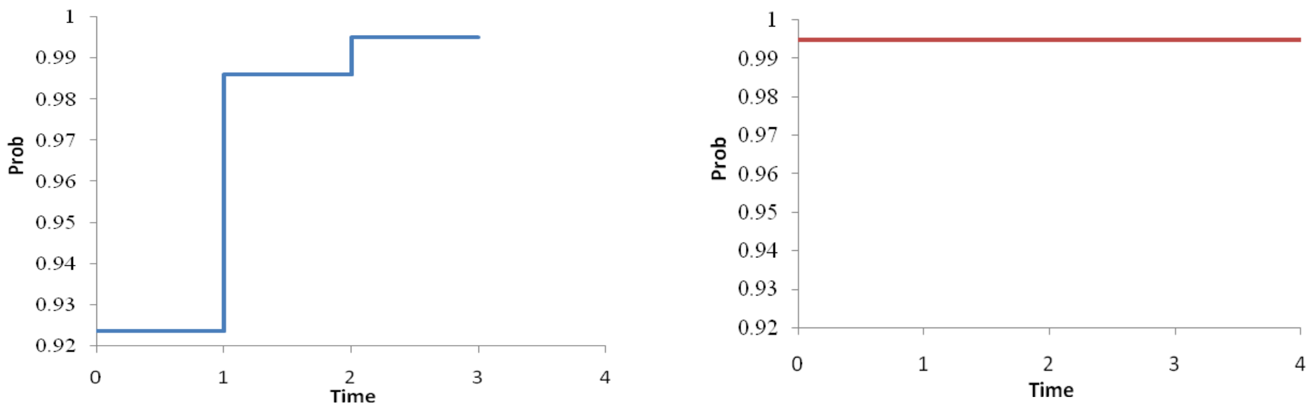
Fig. 4. Temporal Model for the Example TIN

TABLE 2

x_1	$P(z=1\mid x_1^1)$	x_1	x_2	$P(z=1\mid x_1^2)$
0	0.076381	0	0	0.013887
1	**0.923619**	0	1	0.138875
		1	0	0.861125
		1	1	**0.986113**

TABLE 3

x_1	$P_1(B)$	$P(z=1\mid x_1^1)$	x_1	x_2	$P_2(B)$	$P(z=1\mid x_1^2)$	x_1	x_1	x_4	x_3	$P_4(B)$	$P(z=1\mid x_1^4)$
0	0.5	0.005000	0	0	0.5	0.005000	0	0	0	0	0.5	0.005
1		**0.995000**	0	1		0.005000	0	0	0	1		0.005
			1	0		0.995000	0	0	1	0		0.005
			1	1		**0.995000**	0	0	1	1		0.005
							0	1	0	0		0.005
							0	1	0	1		0.005
							0	1	1	0		0.95
							0	1	1	1		0.005
							1	0	0	0		0.005
							1	0	0	1		0.995
							1	0	1	0		0.05
							1	0	1	1		0.995
							1	1	0	0		0.995
							1	1	0	1		0.995
							1	1	1	0		0.995
							1	1	1	1		**0.995**



(a). For Temporal Case I (b). For Temporal Case II

Fig. 5. Probability Profile for the Example COA,

Example 2:

In multi-node connected network structures, given a set of external unaffected affecting events $\{A_i\}_i$, given influence constants $\{h_n(x_1^k)\}_k$, pertinent conditional probabilities are constructed hierarchically, as the structure of the network dictates. Consider, for example, the network in Fig. 6, below. In this network, the affecting events $A_i; i = 1,2,3,4$ are external and unaffected by other events, while events B and C are affected, B being affecting as well. Let us denote the status of event $A_i; i = 1,2,3,4$; by x_i, the status of event B by y and the status of event C by z, where y, z and $\{x_i\}_{1 \le i \le 4}$ are 0-1 binary numbers. Let the influence constants $h(x_1, x_2), h(x_3, x_4)$ and $h(y, x_3, x_4)$ be given. Let also the joint probability $P(x_1, x_2, x_3, x_4)$ be given.

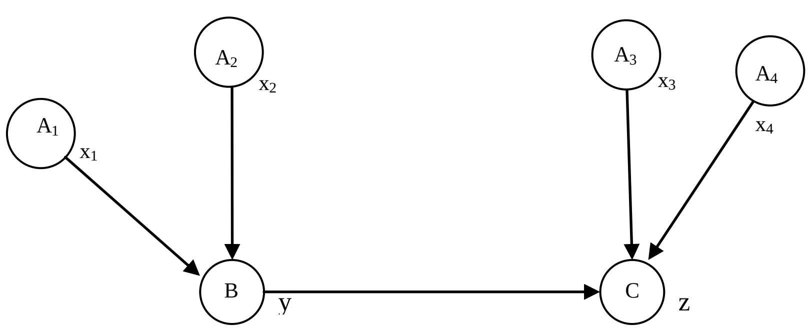


Fig. 6. A Multi-node Network

We then compute all the pertinent probabilities in the above network following the steps stated below:

1. Compute the probability $P(y)$ from the consistency condition:

$$\sum_{x_1, x_2} P(x_1, x_2)\{1 + h(x_1, x_2)[1 - P(y)]P^{-1}(y)\}^{\mathrm{sgn}\, h(x_1, x_2)} \{1 + h(x_1, x_2)\}^{1 - \mathrm{sgn}\, h(x_1, x_2)} = 1$$

where,

$$P(x_1, x_2) = \sum_{x_3, x_4} P(x_1, x_2, x_3, x_4)$$

2. Compute $P(y \mid x_1, x_2)$ from,

$$P(y \mid x_1, x_2) = P(y)\{1 + h(x_1, x_2)[1 - P(y)]P^{-1}(y)\}^{\mathrm{sgn}h(x_1,x_2)}\{1 + h(x_1, x_2)\}^{1-\mathrm{sgn}h(x_1,x_2)}$$

3. Compute $P(y, x_3, x_4)$ as:

$$P(y, x_3, x_4) = \sum_{x_1, x_2} P(y \mid x_1, x_2)P(x_1, x_2, x_3, x_4)$$

4. Compute $P(z)$ from the consistency condition

$$\sum_{y, x_3, x_4} P(y, x_3, x_4)\{1 + h(y, x_3, x_4)[1 - P(z)]P^{-1}(z)\}^{\mathrm{sgn}h(y,x_3,x_4)}\{1 + h(y, x_3, x_4)\}^{1-\mathrm{sgn}h(y,x_3,x_4)} = 1$$

5. Compute $P(z \mid y, x_3, x_4)$ from,

$$P(z \mid y, x_3, x_4) = P(z)\{1 + h(y, x_3, x_4)[1 - P(z)]P^{-1}(z)\}^{\mathrm{sgn}h(y,x_3,x_4)}\{1 + h(y, x_3, x_4)\}^{1-\mathrm{sgn}h(y,x_3,x_4)}$$

6. Compute $P(z \mid x_1, x_2, x_3, x_4)$ from,

$$P(z \mid x_1, x_2, x_3, x_4) = \sum_{y} P(z, y \mid x_1, x_2, x_3, x_4) = \sum_{y} P(z \mid y, x_3, x_4)P(y, \mid x_1, x_2)$$

B. Numerical Evaluations

In this section, we apply the algorithms developed in this paper to an illustrative TIN. We also provide a comparison of the latter results with those previously obtained via the use of the CAST logic. The model used in this section was presented by Wagenhals et al. in 2001 [33] to address the following scenario: As described in [33], internal political instabilities in Indonesia have deteriorated and ethnic tensions between the multiple groups that comprise Indonesia have increased. Religion has been a major factor in these conflicts. Members of one of the minority (2%) religious groups have banded together to combat disenfranchisement. These members have formed a rebel militia group. Armed conflicts recently occurred between those rebels and the Indonesian military. The rebels fled to eastern Java where they have secured an enclave of land. This has resulted in a large number of Indonesian citizens being within the rebel-secured territory. Many of these people are unsympathetic to the rebels and are considered to be at risk. It is feared that they may be used as hostages if ongoing negotiations break down with the Indonesian government. The food and water supply and sanitation facilities are very limited within the rebel-secured territory.

Several humanitarian assistance (HA) organizations are on the island, having been involved with food distribution and the delivery of public health services to the urban poor for several years. So far, the rebels have not prevented HA personnel from entering the territory to take supplies to the citizens. The U.S. and Australian embassies in Jakarta are closely monitoring the situation for any indications of increasing rebel activity. In addition, Thailand, which has sent several hundred citizens to staff numerous capital investment projects on Java, is known to be closely monitoring the situation. To reflect the situation stated above, a TIN was first created in [33] and is shown in Fig. 7.

The latter TIN models the causal and influencing relationships between (external) affecting events (on the left side and along the top of the model in Fig. 7) and the overall effect of concern which is the single node with no parents on the right-hand side of the model. In this case, the effect is “Rebels decide to avoid violence”. The actionable (external) events in this model include a combination of potential coalition, UN, and rebel actions. The coalition actions include actions by the US government, its military instrument of national power, actions by the Government of Indonesia, and actions by Thailand.

For purposes of illustration and comparison of results, we have selected a part of this network, as shown in Fig. 8.

The (external) affecting events in the TIN of Fig. 8 are drawn as root nodes (nodes without incoming edges). The text in each node, e.g., "1—Coalition Deploys Forces to Indonesia," represents a node ID and a statement describing the binary proposition. In Fig. 8, $\{A_i\}_{0 \le i \le 4}$ represents the set of the external affecting events, where the index 'i' depicts the node ID. The marginal probabilities for the external affecting events are also shown inside each node. In this illustration, we assume all external affecting events to be mutually independent (Section IV.) A desired effect, or an objective which a decision maker is interested in, is modeled as a leaf node (node without outgoing edges). The node with ID '10' in Fig. 8 represents the objective for the illustration. In both Figs. 7 and 8, the root nodes are drawn as rectangles while the non-root nodes are drawn as rounded rectangles. A directed edge with an arrowhead between two nodes shows the parent node promoting the chances of a child node being true, while the roundhead edge shows the parent node inhibiting the chances of a child node being true. The first two elements in the inscription associated with each arc quantify the corresponding strengths of the influence of a parent node's state (as being either true or false) on its child node. The third element in the inscription depicts the time it takes for a parent node to influence a child node. For instance, in Fig. 8, event "1—Coalition Deploys Forces to Indonesia" influences the occurrence of event "7—Coalition Secures APOD and SPOD" after 3 time units.

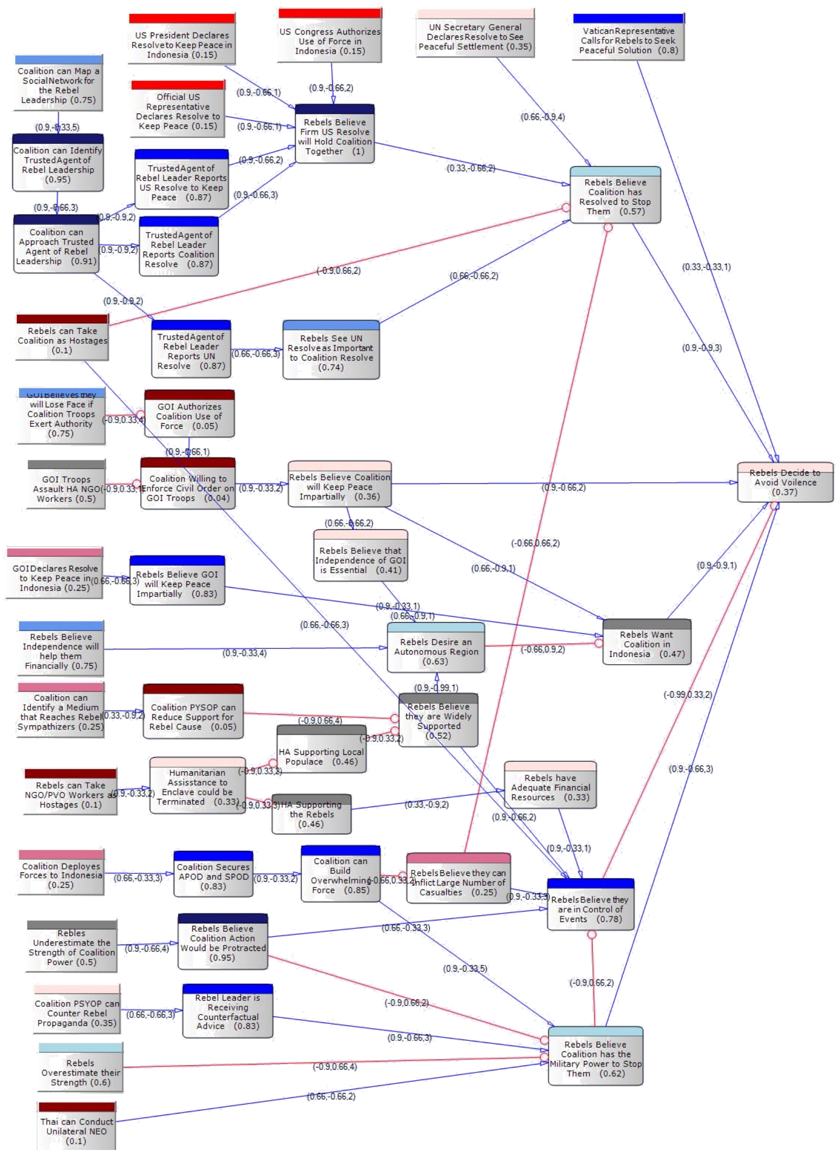


Fig. 7. Timed Influence Net of East Timor Situation [33]

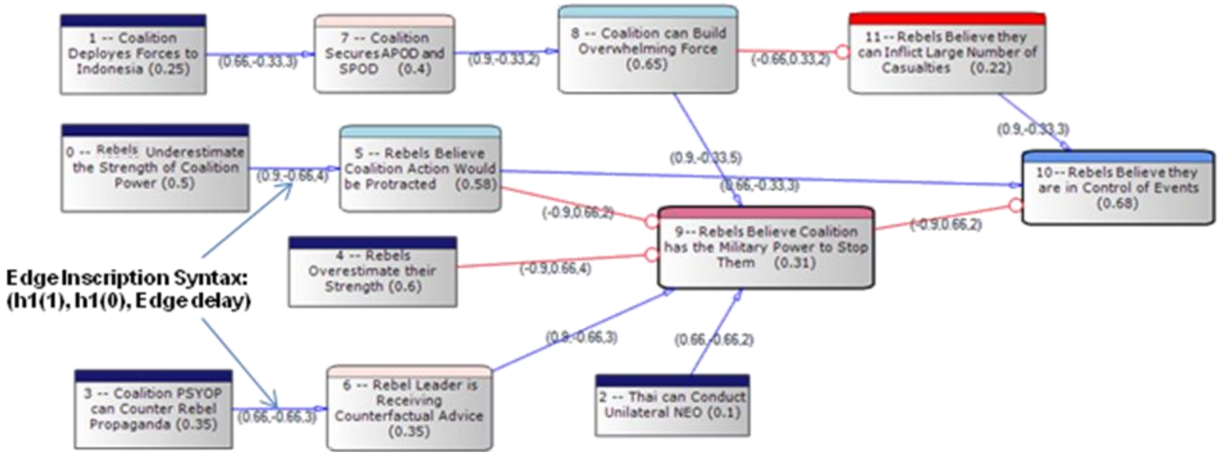


Fig. 8. Sample TIN for Analysis

The purpose of building a TIN is to evaluate and compare the performances of alternative courses of actions described by the set $\boldsymbol{A_T}$ *in the definition of TINs.* The impact of a selected course of action on the desired effect is analyzed with the help of a *probability profile*. The following is an illustration of such an analysis with the help of two COAs, given below:

COA1: *All external affecting events are taken simultaneously at time 1 and are mutually independent.*

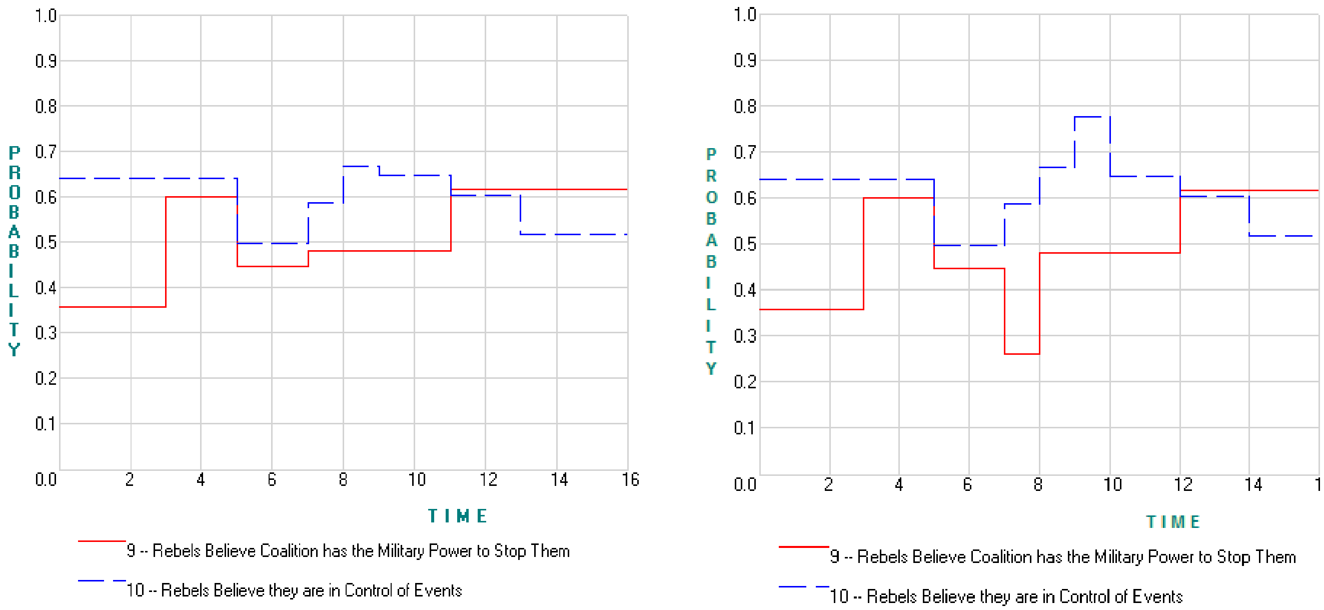
COA2: *Events {0, 2, 4} are taken at time 1, simultaneously, and events {1, 3} are taken at time 2, simultaneously.*

The two COAs can also be described as in Table 4.

Table 4

Event	COA1		COA2	
	Time	**Status**	**Time**	**Status**
0 -- Rebels Underestimate the Strength of Coalition Power	1	1 (ie, True)	1	1
1 -- Coalition Deployes Forces to Indonesia	1	1	2	1
2 -- Thai can Conduct Unilateral NEO	1	1	1	1
3 -- Coalition PSYOP can Counter Rebel Propaganda	1	1	2	1
4 -- Rebels Overestimate their Strength	1	1	1	1

Note that the simultaneous occurrence of external affecting events does not necessarily imply simultaneous revealing of their status on an affected node; the time sequence of revealed affecting events is determined by both the time stamp on each affecting event and the delays on edges. Because of the propagation delay associated with each edge, influences of actions impact the affected event progressively in time. As a result, the probability of the affected event changes as time evolves. A probability profile draws these probabilities against the corresponding time line. In Fig. 9, probability profiles generated for nodes "9—Rebels Believe Coalition has the Military Power to Stop Them" and "10—Rebels Believe they are in Control of Events," using the CAST logic based approach in [2, 22, 28, 29, 34] are shown.



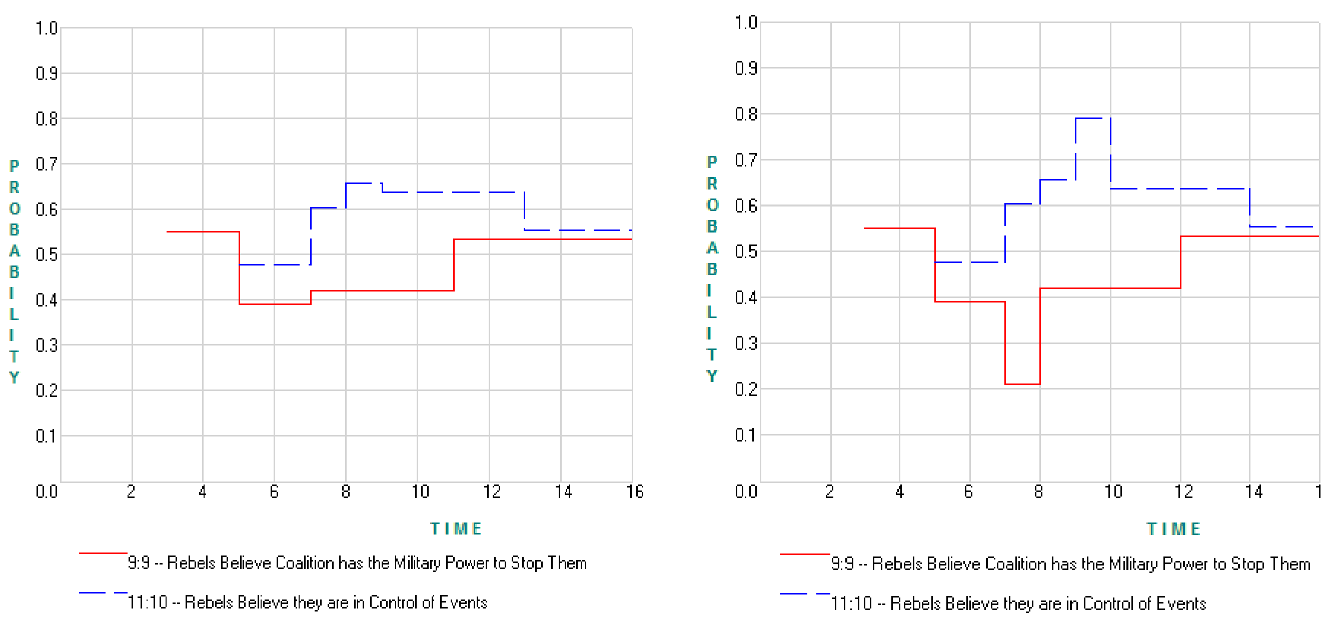
COA1 COA2

Fig. 9. Probability Profiles Generated by the CAST Logic Approach

For the same TIN model as in Fig. 9 and the corresponding course of actions, we used the approach presented in this paper and produced pertinent results for the following two cases:

Case I

For this illustration, we utilize the influence constant model presented in Section IX.A and the temporal case presented in Section VII. The influence constants $\{h_i(x_1^n)\}_{1 \le i \le n-1}$ are first pre-computed via the dynamic programming expression in Lemma 2, Section IV. The resulting probability profiles for the two affected events/propositions in the TIN are shown in Fig. 10.



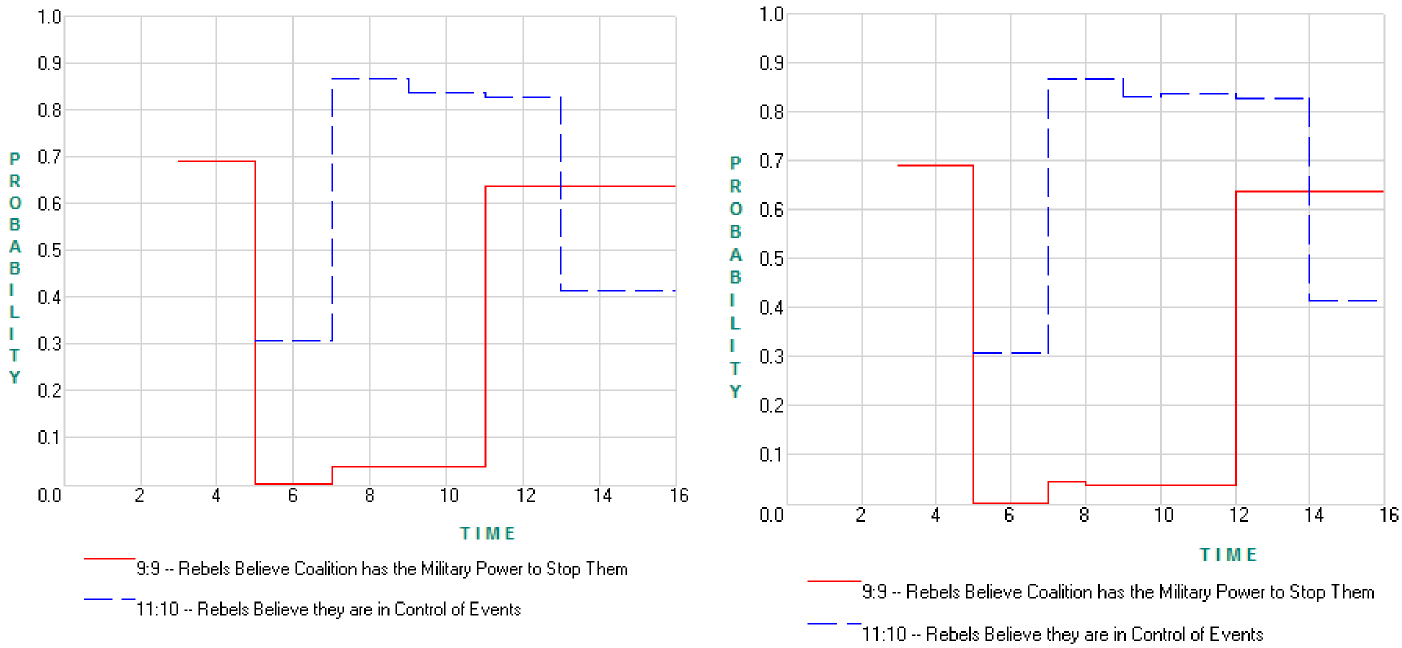
COA1 COA2

Fig. 10. Probability Profiles for Case I

Case II

For this illustration, we utilize the influence constant model presented in Section IX.A and the temporal case presented in

Section IX. In this case, the existence of an affecting event is assumed unknown to an affected event unless it reveals itself and makes its status known to the affected event. The conditional probabilities, in this case, are computed real-time by eq (33). The resulting probability profiles for the two affected events/propositions in the TIN are shown in Fig. 11.



COA1 COA2

Fig. 11. Probability Profiles for Case II

Comparing Figs. 9 and 10, we note that when the existence of all the external affecting events are initially known, then the approach in this paper produces results that are more accurate and consistent than those produced by the CAST logic based approach. This was expected, since the present approach has eliminated the inconsistencies that the CAST logic based approach suffers from. Unlike the CAST logic based approach, the probability profiles generated by the new approach only record the posterior probabilities resulting from the impacts of the external affecting events and do not assume any default initial values; in profiles of Figs. 10 and 11 the first impact is recorded at time '3'. Comparing Figs. 10 and 11, we note that, as expected, when the existence of the external affecting events are revealed sequentially in time then, there is a relatively high level of instability in time evolution, as compared to the case where the existence of all the external affecting events is initially known. The selection of a influence constant and of temporal models for a TIN under construction/analysis is a design issue and is reflected by the differences in the resulting probability profiles.

XII. CONCLUSION

In this paper, we presented a comprehensive approach to Influence Nets including conditions for model consistency and dynamic programming evolution of the influence constants, as well as temporal issues and model testing methodologies. We revisited the earlier CAST logic [2, 22] based approach to Timed Influence Network (TIN) modeling [28, 29, 34], by redefining the design parameters for a TIN model, reevaluating the cases of independence and (partial) dependence among external affecting events, introducing new methods for aggregating joint influences from design parameters, and by offering new insights into the temporal

aspects of causal influences modeled inside a TIN. The presented approach successfully overcomes the deficiencies in the CAST logic based TIN modeling and the inconsistencies therein. It also does not require any additional design information than that already available in a TIN constructed via CAST logic parameters; the entire repository of situational models developed earlier [29, 33, 34] may be simply reanalyzed (without any modifications) using the new set of computational tools introduced in this paper. We analyzed and evaluated our approach and tested it for a specific TIN. The approach produces consistent and stable in time results. It can be effectively utilized to model, test and evaluate large organizations of interest.

References

[1] J. M Agosta, "Conditional Inter-Causally independent Node Distributions, a Property of Noisy-or Models," in *Proceedings of the 7th Conference on Uncertainty in Artificial intelligence*, 1991.

[2] K. C. Chang, P. E. Lehner, A. H. Levis, A. K. Zaidi, and X.Zhao, *On Causal Influence Logic*. Technical Report, George Mason University, Center of Excellence for C31, Dec. 1994.

[3] G. F. Cooper, "The Computation Complexity of Probabilistic Inference Using Bayesian Belief Networks," *Artifiical Intelligence*, 42(2-3): 393-405, March, 1990.

[4] M. J. Drudzel, and M. Henrion, "Intercausal Reasoning with Uninstantiated Ancestor Nodes," in *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, 1993.

[5] S. Haider, "On Computing Marginal Probability Intervals in Probabilistic Inference Networks," in *Proceedings of IEEE-SMC Conference*, October, 2003.

[6] S. Haider, "A Hybrid Approach for Learning Parameters of Probabilistic Networks from Incomplete Databases," *Design and Application of Hybrid Intelligent Systems, Proceedings of the Third International Conference on Hybrid Intelligent Systems (HIS'03)*, IOS Press, Amsterdam, The Netherlands, ISBN 158603-394-8, 2003.

[7] S. Haider, A. K. Zaidi, and A. H. Levis, "A Heuristic Approach for Best Set of Actions Determination in Influence Nets," in *Proc. IEEE International Conference on Information Reuse and Integration*, Las Vegas, Nov. 2004.

[8] S. Haider and A. H. Levis, "An Approximation Technique for Belief Revision in Timed Influence Nets," in *Proceedings of Command and Control Research and Technology Symposium*, 2004.

[9] S. Haider and A. K. Zaidi, "Transforming Timed Influence Nets into Time Sliced Bayesian Networks," in *Proceedings of Command and Control Research and Technology Symposium*, 2004.

[10] S. Haider and A. H. Levis, "Dynamic Influence Nets: An Extension of Timed Influence Nets for Modeling Dynamic Uncertain Situations," in *Proc. 10th International Command and Control Research and Technology Symposium*, Washington DC, June 2005.

[11] S. Haider and A. K. Zaidi, "On Temporal Analysis of Timed Influence Nets using Point Graphs," in *Proc. of the 18th International FLAIRS Conference*, FL, 2005.

[12] S. Haider and A. H. Levis, "Effective Courses of Action Determination to Achieve Desired Effects" in *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 37, No. 6, November 2007.

[13] R. Howard and J. E. Matheson, "Influence diagrams," in *Readings on the Principles and Applications of Decision Analysis*, vol. 2, R. Howard and J. Matheson, Eds. Menlo Park, CA: Strategic Decisions Group, 1981, pp. 721–762.

[14] R. A. Howard and J. E. Matheson, "Influence diagram retrospective," *Decis. Anal.*, vol. 2, no. 3, pp. 144–147, Sep. 2005.

[15] R. A. Howard, J. E. Matheson, M. W. Merkhofer, A. C. Miller, and D. W. North, "Comment on influence diagram retrospective," Decis. Anal., vol. 3, no. 2, pp. 117–119, Jun. 2006.

[16] F.R. Kschischang and B.J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE Journal on Selected Areas in Communication*, 16:219-230, 1998.

[17] R. J. McEliece, McKay D.J.C., and Cheng J.F., "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE Journal on Selected Areas in Communication*, 16(2):140--152, February 1998.

[18] K.P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: an empirical study," in *Proceedings of Uncertainty in AI*, 1999.

[19] Neapolitan, R. E., *Learning Bayesian Networks*, Prentice Hall, 2003.

[20] P. Papantoni-Kazakos, A. K. Zaidi and M. Faraz Rafi, "An Algorithm for Activation Timed Influence Nets" Submitted in *the 2008 IEEE International Conference on Information Reuse and Integration*, Los Vegas, 2008.

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[22] J. A. Rosen and W. L. Smith, "Influence Net Modeling with Causal Strengths: An Evolutionary Approach," in *Proceedings of the Command and Control Research and Technology Symposium*, Naval Post Graduate School, Monterey CA, June 25-28, 1996.

[23] J. A. Rosen and W. L. Smith, "Influencing Global Situations: A Collaborative Approach," *US Air Force Air Chronicles*, 1996.

[24] J. A. Rosen, W. L. Smith, E. S. Smith, and M. A. Maldony, "Planning with Influence Net Modeling: An Architecture for Distributed, Real-Time Collaboration," *Military Operations Research Symposium* (MORS), 1998.

[25] J. A. Rosen and W. L. Smith, "Influence Net Modeling for Strategic Planning: A Structured Approach for Information Operations," *Phalanx*, vol. 33, 2000.

[26] R. D. Shachter, "Evaluating influence diagrams," *Oper. Res.*, vol. 34, no. 6, pp. 871–882, Nov./Dec. 1986.

[27] L. W. Wagenhals, "Course of Action Development and Evaluation Using Discrete Event System Models of Influence Nets", PhD Dissertation, George Mason University, 2000.

[28] L. W. Wagenhals, A. H. Levis, "Course of Action Development and Evaluation," in *Proceedings of the 2000 Command and Control Research and Technology Symposium*, 2000.

[29] L. W. Wagenhals and A. H. Levis, "Modeling Effects Based Operations in Support of War Games," in *Proc. Of the 15th International Symposium on Aerospace / Defense Sensing, Internal Society for Optical Engineering*, Proceedings of SPIE, Vol. # 4367, 2001.

[30] L. W. Wagenhals and A. H. Levis, "Modeling Support of Effects-Based Operations in War Games," in *Proc. 2002 Command and Control Research and Technology Symposium*, Monterey, CA, June 2002.

[31] L. W. Wagenhals, A. H. Levis, and M. McCrabb, "Effects Based Operations: a Historical Perspective for a Way Ahead," in *Proc. 8th Int'l Command and Control Research and Technology Symposium*, National Defense University, Washington, DC, June 2003.

[32] L. W. Wagenhals, I. Shin, and A. H. Levis, "Creating Executable Models of Influence Nets with Coloured Petri Nets," *Int. J. STTT*, Spring-Verlag, Vol. 1998, No. 2, 1998.

[33] L. W. Wagenhals, T. J. Reid, R. J. Smillie, & A. H. Levis, "Course of Action Analysis for Coalition Operations," in *Proceedings of 6th International Command and Control Research and Technology Symposium*, Annapolis, Maryland, Jun. 2001.

[34] L. W. Wagenhals and L. K. Wentz, "New Effects-Based Operations Models in War Games," in *Proceedings of the 2003 International Command and Control Research and Technology Symposium*, National Defense University, Washington, DC, June 2003.

[35] A. K. Zaidi, L W. Wagenhals, and S. Haider, "Assessment of Effects Based Operations Using Temporal Logic," in *Proc. of the 10th International Command and Control Research and Technology Symposium*, 2005.

APPENDIX

A. Proof of Lemma 2

Applying the Bayes Rule, we write

$$P(B \mid x_1^{n-1}) = \frac{P(x_1^{n-1} \mid B)P(B)}{P(x_1^{n-1})} \tag{A.1}$$

Where, due to the Theorem of Total Probability, we have

$$P(x_1^{n-1} \mid B) = \sum_{x_n=0,1} P(x_1^n \mid B) \tag{A.2}$$

From the Bayes Rule we also have,

$$P(x_1^n \mid B) = \frac{P(B \mid x_1^n)P(x_1^n)}{P(B)} \tag{A.3}$$

Substituting (A.2) and (A.3) in (A.1), we obtain:

$$P(B \mid x_1^{n-1}) = \sum_{x_n=0,1} P(B \mid x_1^n)P(x_1^n)P^{-1}(x_1^{n-1}) = \sum_{x_n=0,1} P(B \mid x_1^n)P(x_n \mid x_1^{n-1}) \tag{A.4}$$

Where in (A.4) we used the definition of conditional probability $P(x_n \mid x_1^{n-1}) = P(x_1^n)P^{-1}(x_1^{n-1})$.
Using expression (4) in Section III and substituting in (A.4), we obtain.

$$\begin{aligned}&\{1+h_{n-1}(x_1^{n-1})[1-P(B)]P^{-1}(B)\}^{\operatorname{sgn} h_{n-1}(x_1^{n-1})} \bullet \{1+h_{n-1}(x_1^{n-1})\}^{1-\operatorname{sgn} h_{n-1}(x_1^{n-1})} = \\ &\sum_{x_n=0,1} P(x_n \mid x_1^{n-1})\{1+h_n(x_1^n)[1-P(B)]P^{-1}(B)\}^{\operatorname{sgn} h_n(x_1^n)} \bullet \{1+h_n(x_1^n)\}^{1-\operatorname{sgn} h_n(x_1^n)} \overset{\Delta}{=} Q_n + 1\end{aligned} \tag{A.5}$$

Observing (A.5), we notice that if $(Q_n + 1) \in [0,1]$, then $h_{n-1}(x_1^{n-1})$ must be necessarily negative, reducing the left part of the equality in (A.5) to $1 + h_{n-1}(x_1^{n-1})$. If, on the other hand, $(Q_n + 1) \in [1, P^{-1}(B)]$ then the left part of (A.5) must necessarily reduce to $1 + h_{n-1}(x_1^{n-1})[1 - P(B)]P^{-1}(B)$, with $h_{n-1}(x_1^{n-1})$ positive. The above observations clearly lead to the result in the lemma.

B. Proof of Lemma 3

Due to the Bayes Rule, we have:

$$P(B \mid x_1^n) = \frac{P(x_1^n \mid B)P(B)}{P(x_1^n)} \tag{A.6}$$

Due to the independence assumption, we have:

$$\frac{P(x_1^n \mid B)}{P(x_1^n)} = \prod_{i=1}^{n} \frac{P(x_i \mid B)}{P(x_i)} \tag{A.7}$$

where

$$\frac{P(x_i \mid B)}{P(x_i)} = \frac{P(B \mid x_i)}{P(B)} \tag{A.8}$$

Substituting (A.8) in (A.7) and then (A.6), we obtain:

$$P(B \mid x_1^n) = P^{-(n-1)}(B)\prod_{i=1}^{n} P(B \mid x_i) \tag{A.9}$$

Substituting expression (10) in (A.9) we obtain the expression in the lemma.

C. Proof of lemma 4

Due to the Bayes Rule, we have:

$$\frac{P(B\,|\,x_1^n)}{P(B)} = \frac{P(x_1^n\,|\,B)}{P(x_1^n)} \tag{A.10}$$

Then, due to (13), we obtain from (A.10),

$$\frac{P(B\,|\,x_1^n)}{P(B)} = \frac{P(x_1\,|\,B)}{P(x_1)}\prod_{i=2}^{n}\frac{P(x_i\,|\,x_{i-1},B)}{P(x_i\,|\,x_{i-1})} \tag{A.11}$$

Applying the Bayes Chain Rule, we have:

$$P(x_i\,|\,x_{i-1},B)P(x_{i-1}\,|\,B)P(B) = P(B\,|\,x_i,x_{i-1})P(x_i\,|\,x_{i-1})P(x_{i-1}) \tag{A.12}$$

where,

$$P(x_{i-1}\,|\,B)P(B) = P(B\,|\,x_{i-1})P(x_{i-1}) \tag{A.13}$$

Substituting (A.13) in (A.12) we then obtain:

$$\frac{P(x_i\,|\,x_{i-1},B)}{P(x_i\,|\,x_{i-1})} = \frac{P(B\,|\,x_i,x_{i-1})}{P(B\,|\,x_{i-1})};i\ge 2$$

$$\frac{P(x_1\,|\,B)}{P(x_1)} = \frac{P(B\,|\,x_1)}{P(B)} \tag{A.14}$$

Where, directly from the results in Lemma 1, we have:

$$P(B\,|\,x_i) = \left\{1+h_1^{(i)}(x_i)[1-P(B)]P^{-1}(B)\right\}^{\operatorname{sgn}h_1^{(i)}(x_i)}\left\{1+h_1^{(i)}(x_i)\right\}^{1-\operatorname{sgn}h_1^{(i)}(x_i)} \tag{A.15}$$

where

$$\underset{i\ge 2}{h_1^{(i)}(x_i)} = \begin{cases} Q_{i,i+1}-1 & ;\quad \text{if } Q_{i,i+1}\in[0,1] \\ P(B)[1-P(B)]^{-1}[Q_{i,i+1}-1] & ;\quad \text{if } Q_{i,i+1}\in[1,P^{-1}(B)] \end{cases} \tag{A.16}$$

$$Q_{i,i+1} \overset{\Delta}{=} \sum_{x_{i+1}=0,1} P(x_{i+1}\,|\,x_i)\left\{1+h_2^{(i,i+1)}(x_i,x_{i+1})[1-P(B)]P^{-1}(B)\right\}^{\operatorname{sgn}h_2^{(i,i+1)}(x_i,x_{i+1})}\bullet\left\{1+h_2^{(i,i+1)}(x_i,x_{i+1})\right\}^{1-\operatorname{sgn}h_2^{(i,i+1)}(x_i,x_{i+1})} \tag{A.17}$$

Substitution of expression (A.14) and (A.17) in (A.11), in conjunction with expression (14), give the result in the lemma.

D. Proof of Lemma 5

Due to the Bayes Rule and the Theorem of Total Probability, we have:

$$P(B) = \sum_{x_1^n} P\left(B \mid x_1^n\right) P\left(x_1^n\right) \tag{A.18}$$

Substituting in (A.18) the expression (27) for the conditional probability $P\left(B \mid x_1^n\right)$, we obtain:

$$\begin{aligned} &\sum_{x_1^n : \sum_{i=1}^n h_1(x_i) > 0} P\left(x_1^n\right)\left\{1 + P^{-1}(B)[1 - P(B)] \max_{1 \le i \le n} h_1(x_i)\right\} + \\ &\sum_{x_1^n : \sum_{i=1}^n h_1(\backslash x_i) = 0} P\left(x_1^n\right) + \sum_{x_1^n : \sum_{i=1}^n h_1(x_i) < 0} P\left(x_1^n\right)\left\{1 + \min_{1 \le i \le n} h_1(x_i)\right\} = 1 \end{aligned} \tag{A.19}$$

which gives after simplification:

$$[1 - P(B)] \sum_{x_1^n : \sum_{i=1}^n h_1(x_i) > 0} P\left(x_1^n\right) \max_{1 \le i \le n} h_1(x_i) + P(B) \sum_{x_1^n : \sum_{i=1}^n h_1(x_i) < 0} P\left(x_1^n\right) \min_{1 \le i \le n} h_1(x_i) = 0 \tag{A.20}$$

An Algorithm for Activation Timed Influence Nets*

P. Papantoni-Kazakos, *Fellow, IEEE*
Department of Electrical Engineering
University of Colorado Denver, CO
Titsa.Papantoni@cudenver.edu

Abbas K. Zaidi, *SM, IEEE,* M. Faraz Rafi
System Architectures Lab
George Mason University, Fairfax, VA
szaidi@ieee.org, *mrafi1@gmu.edu*

Abstract

Activation Timed Influence Net (ATIN) is a term representing a progressively evolving sequence of actions, where the effects of an action become the preconditions of the action that follows. An ATIN integrates the notions of time and uncertainty in a network model, where nodes explicitly represent mechanisms and/or tactical actions that are responsible for changes in the state of a domain. In this paper, we present an algorithm for the initialization of actions within a ATIN.

1. Introduction

We consider the scenario where a sequence of actions needs to be initialized towards the materializing of some desirable effects. As depicted in Figure 1, each action is supported by a set of preconditions and gives rise to a set of effects; the latter become then the preconditions of the following action(s) which, in turn, gives rise to another set of effects. Such sequential evolution of actions is termed as Activation Timed Influence Nets (ATINs), where the action performers may be humans. ATINs are an extension of an earlier formalism called Timed Influence Nets (TINs) [2-18] that integrate the notions of time and uncertainty in a network model. The TIN's are comprised of nodes that represent propositions (i.e., pre-and post-conditions of potential actions as well as assertions of events which may indirectly describe such actions), connected via causal links that represent relationships between the nodes, without any explicit representation of actions. TINs have been experimentally used in the area of Effects Based Operations (EBOs) for evaluating alternate courses of actions and their effectiveness to mission objectives in a variety of domains, e.g., war games [9-11, 14], and coalition peace operations [13, 16], to name a few. A number of analytical tools [2, 4-8, 12-13, 15, 17] have also been developed over the years for TIN models to help an analyst update conditions/assertions, represented as nodes in a TIN, to map a TIN model to a Time Sliced Bayesian Network for incorporating feedback evidence, to determine best set of pre-conditions for both timed and un-timed versions of Influence Nets, and to assess temporal aspects of the influences between nodes. A recent work [18] on TINs, underlying constructs and the computational algorithms, provides a comprehensive analytical underpinning of the modeling and analysis approach.

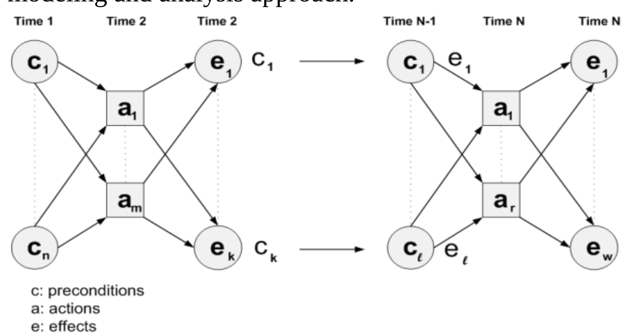


Figure 1. Network Representation of an Activation Timed Influence Net (ATIN)

In contrast to their predecessors (TINs), ATINs explicitly incorporate as nodes the mechanisms and/or actions that are responsible for changes in the state of a domain; other nodes represent preconditions and effects of actions. A set of preconditions may support a number of different actions, each of which may lead to the same effects, with different probabilities and different costs/awards, however. The objective is to select an optimal set of actions, where optimality is determined via a pre-selected performance criterion. In this paper, we present two algorithms which attain such an objective. We note that an effort to develop an action selection algorithm is also presented in [1].

2. Problem Formalization – Core

For clarity in presentation, let us initially isolate a single action with its supporting preconditions and its resulting effects, as depicted in Figure 2.

* This work was supported by the Air Force Office of Scientific Research (AFOSR) under Grants FA9550-05-1-0106 and FA9550-05-1-0388.

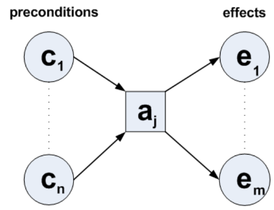


Figure 2. A Single Action ATIN

Let us denote,

$X_1^n = [X_1,..., X_n]^T$	The status random vector of the preconditions, where $X_i = 1$, if precondition c_i is present and $X_i = 0$ if precondition c_i is absent. x_1^n denote binary vector realizations of X_1^n.
$Y_1^m = [Y_1,..., Y_m]^T$	The status random vector of the effects, where $Y_i = 1$, if effect e_i is present and $Y_i = 0$ if effect e_i is absent. y_1^m denote binary vector realizations of Y_1^m.
$p_j(x_1^n)$	The probability of success for action a_j, given that the value of the precondition status vector is x_1^n; $P(\text{success for action } a_j \mid x_1^n)$
$q_j(y_1^m)$	The probability that the value of the effects' status vector is y_1^m, given that the action a_j is taken; $P(y_1^m \mid a_j \text{ taken})$
$q_0(y_1^m)$	The probability that the value of the effects' status vector is y_1^m, given that no action is taken; $P(y_1^m \mid \text{no action taken})$
$U_j(y_1^m)$	The utility of the value y_1^m of the effects' status vector, when action a_j is taken.
$U_0(y_1^m)$	The utility of the value y_1^m of the effects' status vector, when no action is taken.

We note that the utility function $U_j(y_1^m)$ measures the net worth of the effects' vector value y_1^m, when action a_j is taken; thus, $U_j(y_1^m)$ is computed as the worth of y_1^m minus the cost of deployment for action a_j.

Let us now assume mutually exclusive actions a_1 to a_k, which are supported by the same preconditions c_1 to c_n, to lead to the same set of effects e_1 to e_m (as shown in Figure 3). Let $\{a_j\}$ $1 \le j \le k$ be this set of actions and let X_1^n and Y_1^m denote the common status random vectors of preconditions versus effects, respectively. Let the utility functions for each action in the set $\{a_j\}_{1 \le j \le k}$ be nonnegative; let also $U_0(y_1^m)$ be nonnegative.

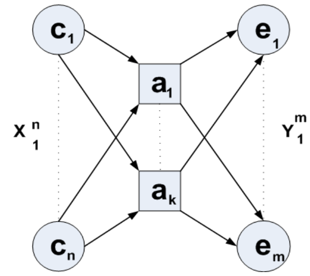


Figure 3. A Single Level ATIN

We now state two different versions of the core problem, based on two different optimization criteria.

Problem 1 (Optimal Path Problem)

Given a preconditions vector value x_1^n, given an effects vector value y_1^m, find the maximum probability action that connects them. That is, find the action that maximizes the conditional probability $P(y_1^m \mid x_1^n)$.

Problem 2 (Average Utility Maximization)

Given a preconditions vector value x_1^n, find the action that maximizes the effects' average utility.

3. Solutions to the Core Problems

We present the solutions to the two core problems posed in Section 2 in the form of a theorem.

<u>Theorem 1</u>

a. Given x_1^n, given y_1^m, and given a set of actions $\{a_j\}_{1 \le j \le k}$, the conditional probability $P(y_1^m \mid x_1^n)$ is maximized as follows:

   by action a_{j*} (where '$j*$' refers to the index of selected action); if

   $$q_{j*}(y_1^m)\, p_{j*}(x_1^n) = \max_{1 \le j \le k} q_j(y_1^m)\, p_j(x_1^n) > q_0(y_1^m) \tag{1}$$

   where then $\max P(y_1^m \mid x_1^n) = q_{j*}(y_1^m)\, p_{j*}(x_1^n)$

   by no action; if

   $$q_0(y_1^m) > \max_{1 \le j \le k} q_j(y_1^m)\, p_j(x_1^n) \tag{2}$$

   where then $\max P(y_1^m \mid x_1^n) = q_0(y_1^m)$

   If more than one actions satisfy the maximum in (1), then one of these actions may be selected randomly.

b. Given X_1^n, given a set of actions $\{a_j\}_{1 \le j \le k}$, and given utility functions $\{U_j(y_1^m)\}_{1 \le j \le k}$ and $U_0(y_1^m)$, the

average utility

$$\overline{U}(x_1^n) \overset{\Delta}{=} \sum_{1\le j\le k} \sum_{y_1^m} P(a_j \text{ taken}, y_1^m \mid x_1^n)\, U_j(y_1^m) + \sum_{y_1^m} P(\text{no action taken} \mid x_1^n)\, U_0(y_1^m)$$

is maximized as follows:

by action a_{j*}; if

$$A_{j*}(x_1^n) \overset{\Delta}{=} p_{j*}(x_1^n) \sum_{y_1^m} q_{j*}(y_1^m)\, U_{j*}(y_1^m) = \max_{1\le j\le k} p_j(x_1^n) \sum_{y_1^m} q_j(y_1^m)\, U_j(y_1^m) > \sum_{y_1^m} q_0(y_1^m)\, U_0(y_1^m) \tag{3}$$

by no action; if

$$\sum_{y_1^m} q_0(y_1^m)\, U_0(y_1^m) > \max_{1\le j\le k} p_j(x_1^n) \sum_{y_1^m} q_j(y_1^m)\, U_j(y_1^m) \tag{4}$$

$A_{j*}(x_1^n)$ in (3) is the award assigned to action a_{j*}; it is also the worth assigned to the precondition vector value x_1^n by the action a_{j*}.

If more than one actions attain the maximum award $A_{j*}(x_1^n)$ in (3), one of them is selected randomly.

4. Solution of the Network Propagation Problem

In this section, we generalize the core solutions expressed in Theorem 1, Section 3, to the sequence of actions depicted by the ATIN in Figure 1, Section 1.

Problem 1 (The Optimal Path Problem)

In the ATIN in Figure 1, we fix the preconditions vector value $x_1^n(1)$, at time 1, and the effects' vector value $y_1^w(N)$, at time N. We then seek for the sequence of actions that maximizes the probability $P(y_1^w(N) \mid x_1^n(1))$. The solution to this problem is of dynamic programming nature, whose steps are as follows, where $x_1^n(l) = y_1^m(l-1)$; $2 \le l \le N$, in our notation.

Step 1: For each $y_1^m(1) = x_1^n(2)$ value, find $r(y_1^m(1)) \overset{\Delta}{=} \max\left(q_0\,(y_1^m(1)),\ \max_j q_j\,(y_1^m(1))\ p_j\,(x_1^n(1))\right)$ and the action index $j^*(y_1^m(1))$ that attains $r(y_1^m(1))$

Step ℓ: The values $r(y_1^m(l\text{-}1)) \overset{\Delta}{=} \max P(y_1^m(l-1) \mid x_1^n(1))$, for each $y_1^m(l-1)$ value, are in memory, as well as the actions that attain them. At step ℓ, the values $r(y_1^m(l)) \overset{\Delta}{=} \max_{y_1^m(\ell-1)} \max\left(\max_j p_j\,(y_1^m(\ell\text{-}1))\, q_j\,(y_1^m(\ell)),\ q_0\,(y_1^m(\ell))\right)$ are maintained, as well as the sequence of actions leading to them.

Problem 2 (The Average Utility Maximization)

In the ATIN in Figure 1, we fix the value of the precondition vector at time 1, denoted $x_1^n(1)$. For each value $y_1^w(N)$ of the effects vector at time N, we assign worth functions $U(y_1^w(N))$. For each action $a_j(l)$, at time l, we assign a deployment cost $c_j(l)$. The utility of the effects' vector value $y_1^w(N)$, when action $a_j(N)$ is taken, is then equal to $U_j(y_1^w(N)) \overset{\Delta}{=} U(y_1^w(N)) - c_j(N)$, while the utility of the same value, when no action is taken, equals $U_0(y_1^w(N)) \overset{\Delta}{=} U(y_1^w(N))$. We are seeking the sequence of actions which lead to the maximization of the average utility. The evolving algorithm, from part (b) of Theorem 1, back propagates as follows:

Step 1: Compute the action awards (including that to no action), with notation of Figure 1, as follows: $0 \le j \le r$;

$$A_j(x_1^\ell(N-1)) \overset{\Delta}{=} p_j(x_1^\ell(N\text{-}1)) \sum_{y_1^w(N)} q_j(y_1^w(N))\, U_j(y_1^w(N))\ ;\ \text{with } p_0(x_1^\ell(N\text{-}1)) \overset{\Delta}{=} 1$$

Select $A_{j*(x_1^\ell(N-1))}(x_1^\ell(N-1)) \overset{\Delta}{=} \max_{0\le j\le r} A_j(x_1^\ell(N-1))$; for each $x_1^\ell(N-1)$ value.

Take action $a_{j*(x_1^\ell(N-1))}(N)$ for preconditions vector value $x_1^\ell(N-1)$ and simultaneously assign worth $A_{j*(x_1^\ell(N-1))}(x_1^\ell(N-1))$ to $x_1^\ell(N-1)$.

That is, assign:

$$U(x_1^\ell(N-1)) = A_{j*(x_1^\ell(N-1))}(x_1^\ell(N-1)) \tag{5}$$

Step 2: Back propagate to the preconditions at N-2, as in Step 1, starting with the worth assignments in (5), and subsequent utilizations

$$U_j(x_1^\ell(N-1)) = \max[A_{j*(x_1^\ell(N-1))}(x_1^\ell(N-1)) - c_j(N-1), 0]$$

Step n: As in Steps 1 and 2 (for subsequent levels.)

The above described algorithm generates the optimal sequence of actions for given initial preconditions $x_1^n(1)$. The optimal such preconditions can be also found via

maximization of the utility $U_j(x_1^k(2))$, with respect to $x_1^n(1)$.

5. Numerical Evaluations

Experimental Setup

Considering the network in Figure 1, assign:

a. Worth function $U(y_1^w(N))$ for all $y_1^w(N)$ values of the effects' status vector, at level N.

b. Probabilities $q_j(x_1^k(l)) \overset{\Delta}{=} P(x_1^k(l)$ occurring | action j at step l-1) at all levels, 2 to N, where $q_0(x_1^k(l)) \overset{\Delta}{=} P\ (x_1^k(l)$ occurring | no action j at step l-1) at all levels, 2 to N,

c. Probabilities $p_j(x_1^k(l)) \overset{\Delta}{=}$ P(action j succeeds | $x_1^k(l)$ preconditions) at all levels, from 1 to N - 1, where $p_0(x_1^k(l)) \overset{\Delta}{=} 1;\ \forall \ell$

d. Implementation/deployment costs $c_j(l)$ for all actions, at all levels 2 to N.

Given the above assignments,

a. Compute first,

$$A_j(x_1^\ell(N-1))$$
$$\overset{\Delta}{=} p_j(x_1^\ell(N-1)) \sum_{y_1^w(N)} q_j(y_1^w(N))\, U_j(y_1^w(N))$$

where, $p_0(x_1^\ell(N-1)) \overset{\Delta}{=} 1$;

$U_j(y_1^w(N)) = \max\,[U(y_1^w(N)) - c_j(N), 0)]$

$$A_{j^*(x_1^\ell(N-1))}(x_1^\ell(N-1)) \overset{\Delta}{=} \max_{0\le j\le r} A_j(x_1^\ell(N-1)) \quad ;$$

For all $x_1^\ell(N-1)$ values.

b. Take action $a_{j^*(x_1^\ell(N-1))}$ for each precondition vector value $x_1^\ell(N-1)$.

Assign worth $A_{j^*(x_1^\ell(N-1))}(x_1^\ell(N-1))$ to $x_1^\ell(N-1)$, as

$U(x_1^\ell(N-1)) = A_{j^*(x_1^\ell(N-1))}(x_1^\ell(N-1))$

c. Repeat steps (a) and (b) for level N-1 and back propagate to level N-2. Continue back propagation to level 1.

6. Application

In this section, we illustrate the use of Activation Timed Influence Networks with the help of an example ATIN, and present the results of the algorithms in this paper when applied to this ATIN. The model used in this section was derived from a TIN presented in Wagenhals *et al.* in 2001 [16] to address the internal political instabilities in Indonesia in the context of East Timor. For purposes of illustration of results, we have selected a part of this network as shown in Fig. 4.

Table 1 lists some of the required parameters and their values of the network in Figure 4. The parameters in this table are listed by their abbreviated labels instead of the phrases shown inside nodes in the figure. (For sake of brevity we do not list all the values.) Only the costs associated with actions a_8 and a_9 in Figure 4 are given in Table 2.

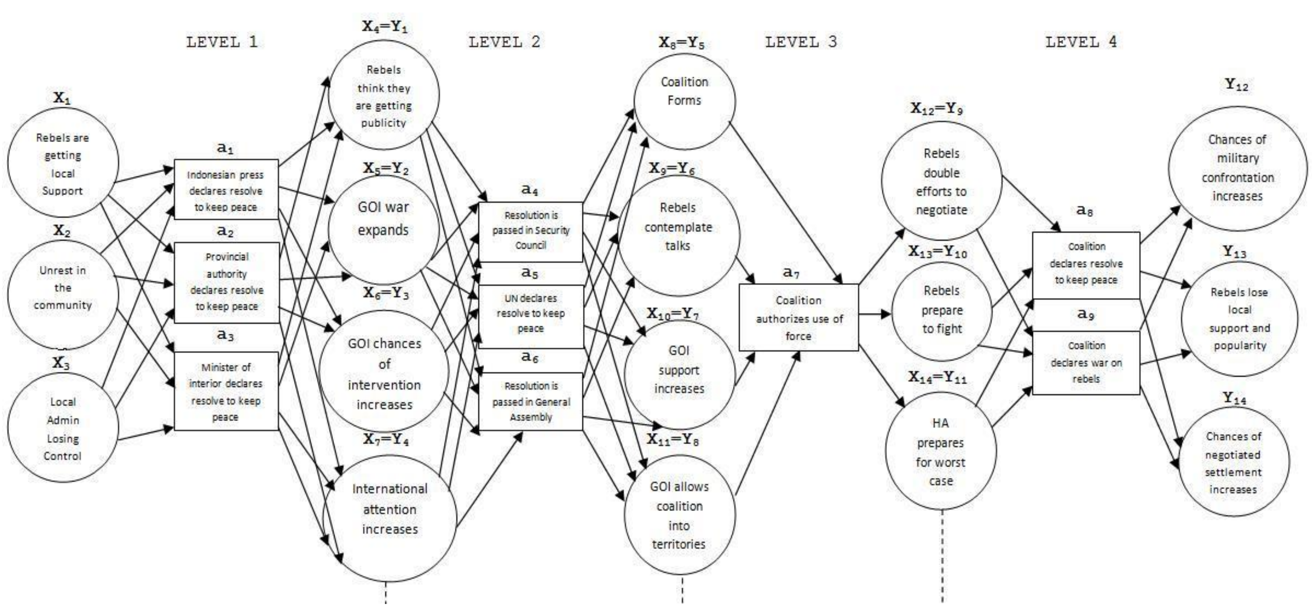


Figure 4. Example ATIN

Table 1. Parameter Values in the Example ATIN

Level 1				
X_1^3	**Action a_j**	**$p_j([1,0,1]^T)$**	Y_1^4	**$q_j(y_1^4)$**
$[1,0,1]^T$	1	0.0140	$[1,1,1,1]^T$	0.888
	2	0.0377	$[0,1,0,1]^T$	0.9029
	3	0.3194	$[1,0,0,0]^T$	0.9988
	0	-	$[1,0,0,0]^T$	0.0074
Level 2				
X_4^7	**Action a_j**	**$p_j([1,0,0,0]^T)$**	Y_5^8	**$q_j(y_5^8)$**
$[1,0,0,0]^T$	4	0.9098	$[1,0,1,0]^T$	0.9538
	5	0.8973	$[0,1,1,1]^T$	0.9897
	6	0.0418	$[1,0,1,0]^T$	0.9528
	0	-	$[0,1,1,1]^T$	0.0812
Level 3				
X_8^{11}	**Action a_j**	**$p_j([0,1,1,1]^T)$**	Y_9^{11}	**$q_j(y_9^{11})$**
$[0,1,1,1]^T$	7	0.5	$[0,0,1]^T$	0.9738
	0	-	$[0,0,1]^T$	0.0742
Level 4				
X_{12}^{14}	**Action a_j**	**$p_j([0,0,1]^T)$**	Y_{12}^{14}	**$q_j(y_{12}^{14})$**
$[0,0,1]^T$	8	0.8383	$[0,1,1]^T$	0.4473
	9	0.3555	$[0,1,1]^T$	0.8016
	0	-	$[0,1,1]^T$	0.0684

Solution to Problem 1:

Consider a scenario in which we need to identify the optimal path (i.e. sequence of actions) which leads to the following output state:

- Reduction in the chances of military confrontation (i.e., $Y_{12} = 0$)
- Loss of local support and popularity by Rebels (i.e., $Y_{13} = 1$)
- Increase in the chances of negotiated settlement (i.e., $Y_{14} = 1$)

The above conditions lead to a post-condition vector $[0, 1, 1]^T$ at level 4, i.e. $y_1^m(N)$.

The precondition vector value (at level 1) is set to be $[1, 0, 1]^T$, where

- $X_1 = 1$ represents the condition, “Rebels are getting local support.”
- $X_2 = 0$ represents the condition, “No unrest in the community.”
- $X_3 = 1$ represents the condition, “Local administration is losing control.”

The sequence of actions that maximizes the probability $P(y_{12}^{14}(4) \mid x_1^3(1))$, obtained as a result of the application of algorithm in Section 4, is given by: **a_3** (i.e., Minister of interior declares resolve to peace), **a_5** (i.e., Sec. Gen. UN declares resolve to keep peace), **a_7** (i.e., Coalition authorizes use of force), **a_8** (i.e., Coalition declares resolve to keep peace).

Only few probability values are shown in Table 1 for the sake of limitation of space, other values are obtained while making use of a complete probability profile not given in this paper.

Solution to Problem 2:

Consider a scenario in which we need to identify the sequence of actions which maximizes the effects' average utility (at level 4) for the same input precondition as was used in Problem 1, i.e. $[1, 0, 1]^T$. The action costs are given in Table 2 whereas; the calculated values of the utility functions $U_j(y_{12}^{14}(4))$ at level 4 are given in Table 3.

Table 2. Action Costs

Action a_j	cost
a_8	25
a_9	30

Table 3. Utility Functions

Y_{12}^{14}	$U(y_{12}^{14}(4))$	$U_8(y_{12}^{14}(4))$	$U_9(y_{12}^{14}(4))$
$[0,0,0]^T$	40	15	10
$[0,0,1]^T$	30	5	0
$[0,1,0]^T$	60	35	30
$[0,1,1]^T$	70	45	40
$[1,0,0]^T$	50	25	20
$[1,0,1]^T$	45	20	15
$[1,1,0]^T$	65	40	35
$[1,1,1]^T$	40	15	10

Table 4. Action Awards

Level 1	Level 2	Level 3	Level 4
$A_2(x_1^3(0))$	$A_4(x_4^7(1))$	$A_7(x_8^{11}(2))$	$A_9(x_{12}^{14}(3))$
838.8	266.8	77.43	79.54

Table 4 summarizes the action awards of those actions which maximize the effects' average utility in their respective levels.

From Table 4 it can be seen that the sequence of actions that maximizes the effects' average utility, obtained as a result of the application of algorithm in Section 4, is given by: **a_2** (i.e., Provincial authority declares resolve to peace),

a_4 (i.e., Resolution is passed in Security Council), **a_7** (i.e., Coalition authorizes use of force), **a_9** (i.e., Coalition declares war on rebels). The resultant set of effects Y_{12}^{14} comes out to be $[1,0,0]^T$, i.e. chances of military confrontation increases, rebels don't lose local support and popularity, and the chances of negotiated settlement decrease.

7. Conclusion

This paper presented an extension of TIN (Timed Influence Net) known as ATIN (Activation Timed Influence Net) which makes use of a set of preconditions required for an action to take place and as a result, produces a set of effects. These effects then become the preconditions for the next level of action(s) and in this way produce a sequential evolution of actions. The paper identified two pre-selected performance criteria regarding ATINs (i.e. optimal path and average utility maximization) and suggested the algorithms to achieve these objectives. The implementation of these algorithms was illustrated with the help of a real world example.

References

[1] Bagchi S., Biswas G., and Kawamura K., "Task Planning under Uncertainty using a Spreading Activation Network", IEEE Transactions on Systems, Man and Cybernetics, Vol. 30, No. 6, November 2000, pp 639-650.

[2] Haider S., "On Computing Marginal Probability Intervals in Probabilistic Inference Networks," in Proceedings of IEEE-SMC Conference, October, 2003.

[3] Haider S. "A Hybrid Approach for Learning Parameters of Probabilistic Networks from Incomplete Databases," Design and Application of Hybrid Intelligent Systems, Proceedings of the Third International Conference on Hybrid Intelligent Systems (HIS'03), IOS Press, Amsterdam, The Netherlands, ISBN 158603-394-8, 2003.

[4] Haider S., Zaidi A. K., and Levis A. H., "A Heuristic Approach for Best Set of Actions Determination in Influence Nets," in Proc. IEEE International Conference on Information Reuse and Integration, Las Vegas, Nov. 2004.

[5] Haider S. and Levis A. H., "An Approximation Technique for Belief Revision in Timed Influence Nets," in Proceedings of Command and Control Research and Technology Symposium, 2004.

[6] Haider S. and Zaidi A. K., "Transforming Timed Influence Nets into Time Sliced Bayesian Networks," in Proceedings of Command and Control Research and Technology Symposium, 2004.

[7] Haider S. and Zaidi A. K., "On Temporal Analysis of Timed Influence Nets using Point Graphs," in the Proc. of the 18th International FLAIRS Conference, FL, 2005.

[8] Haider S. and Levis A. H., "Dynamic Influence Nets: An Extension of Timed Influence Nets for Modeling Dynamic Uncertain Situations," in Proc. 10th International Command and Control Research and Technology Symposium, Washington DC, June 2005.

[9] Wagenhals L. W. and Levis A. H., "Modeling Support of Effects-Based Operations in War Games," in Proc. 2002 Command and Control Research and Technology Symposium, Monterey, CA, June 2002

[10] Wagenhals L.W., Levis A. H., and McCrabb M., "Effects Based Operations: a Historical Perspective for a Way Ahead," in Proc. 8th Int'l Command and Control Research and Technology Symposium, National Defense University, Washington, DC, June 2003.

[11] Wagenhals L. W. and Wentz L. K., "New Effects-Based Operations Models in War Games," in Proceedings of the 2003 International Command and Control Research and Technology Symposium, National Defense University, Washington, DC, June 2003.

[12] Wagenhals L. W., "Course of Action Development and Evaluation Using Discrete Event System Models of Influence Nets", PhD Dissertation, George Mason University, 2000.

[13] Wagenhals L. W., Levis A. H., "Course of Action Development and Evaluation," in Proceedings of the 2000 Command and Control Research and Technology Symposium, 2000.

[14] Wagenhals L. W. and Levis A. H., "Modeling Effects Based Operations in Support of War Games," in Proc. Of the 15th International Symposium on Aerospace / Defense Sensing, Internal Society for Optical Engineering, Proceedings of SPIE, Vol. # 4367, 2001.

[15] Wagenhals L. W., Shin I., and Levis A. H., "Creating Executable Models of Influence Nets with Coloured Petri Nets," Int. J. STTT, Spring-Verlag, Vol. 1998, No. 2, 1998.

[16] Wagenhals L. W., Reid T. J., Smillie R. J., & Levis A. H., "Course of Action Analysis for Coalition Operations," Proceedings of 6th International Command and Control Research and Technology Symposium, Annapolis, Maryland, Jun. 2001.

[17] Zaidi A. K., Wagenhals L W., and Haider S., "Assessment of Effects Based Operations Using Temporal Logic," in Proc. of the 10th International Command and Control Research and Technology Symposium, 2005.

[18] Zaidi A. K., Mansoor F, and Papantoni-Kazakos P., "Modeling with Influence Networks Using Influence Constants: A New Approach," in Proceedings of IEEE-SMC Conference, October, 2007.

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
SYSTEM ARCHITECTURES LABORATORY

PLANNING FOR NETWORK CENTRIC OPERATIONS*

David S. Alberts

September 2005

* This work was supported by the Air Force Office of Scientific Research (AFOSR) under contract no. FA9550-05-1-0106.

Table of Contents

Introduction

The development and maturation of Network Centric Operations is one of the two major dimensions of an Information Age Transformation of the DoD. The other dimension, the mission space, a space that represents the full range of the operations a force must be able to successfully undertaken, is being transformed as well. The 21st century mission space encompasses a wide range of operations including civil-military operations that require (1) an effects-based approach to operations and (2) the ability to work effectively in coalition environments that include not only other militaries but also other government entities, international organizations, and a variety of non-governmental and private voluntary organizations (NGOs and PVOs).

Network Centric Operations require the coevolution of concepts of operation, approaches to command and control (including organization, doctrine, and C2 and information processes) with a robustly networked force, and their materiel and systems. Planning is an integral part of command and control processes, and thus needs to be "reinvented" in order to leverage the capabilities of a robustly networked force and be compatible with network-centric concepts of operation. Thus, moving to Network Centric Operations involves a redefinition of command arrangements and processes, including the adoption of effects-based planning, better integration of planning and execution, and a redefinition of the nature of mission participants and their respective roles, responsibilities, and interactions.

Transformation is by definition more than incremental improvements or sustaining innovations. Transformation requires venturing beyond one's comfort zones to explore new concepts of operation, new approaches to command and control, and new processes. As such, it would be unreasonable to expect the answers to be apparent or the data for analysis to be available. The way ahead involves the formulation, design, and implementation of a campaign of experimentation and an associated program of research focused on the development and assessment of interactive and dynamic effects-based planning in the context of 21st century Network Centric Operations.

Research Scope and Objectives

This research effort found that there was an urgent need for a campaign of research and experimentation focused on developing a network-centric approach to air and space command and control, specifically the development and assessment of approaches to mission planning in a network-centric environment. Having concluded that such a campaign of experimentation is necessary, this document provides the intellectual foundation for such a campaign. It provides appropriate definitions for key concepts, a conceptual reference model for network-centric, effects-based planning and execution, identifies a set of research issues, and identifies key activities that are on the critical path to transforming the planning and execution of air and space operations.

Sponsorship

This research effort was sponsored by the Air Force Office of Scientific Research (AFOSR) under contract no. FA9550-05-1-0106 and performed at the Systems Architectures Laboratory, Department of Electrical and Computer Engineering, George Mason University.

Part One: Understanding NCO

Key Concepts

Despite the fact that key terms associated with Information Age Transformation, such as "Network Centric Warfare," "Network Centric Operations," and "Effects-Based Operations" are in widespread use and that many profess to understand these concepts, these terms are often used incorrectly. Traditional concepts such as "Command and Control" and "Planning" are in the process of being reinvented and redefined. Although they have official definitions, these no longer make sense in light of the dramatic changes that have taken place in both the nature of the missions to be undertaken and the opportunities afforded by advancing technology, particularly information technologies, to do things differently.

However, these terms must be adequately defined and understood to enable the conceptual model presented later in this document to be effectively employed to formulate and guide a successful campaign of experimentation and to fully leverage the data collected and the results of the analyses that are part of the campaign. For this reason, the meanings of key terms are discussed here and citations are provided for readers who would like to have more detailed treatments of these concepts.

Network Centric Warfare / Operations

Network Centric Warfare (NCW), because it is "no less than the embodiment of an Information Age transformation of the DoD,"[1] defies a bumper-sticker definition. The term Network Centric Operations (NCO) was introduced to emphasize that the principles of NCW are applicable to operations of all sorts.

While it is easy to make simple statements that are true about NCW (e.g., it is about a robustly networked force; it is about leveraging Information Age technologies), these simple statements fail to adequately convey the intended meaning and as often as not lead to misunderstandings regarding the scope and implications of NCW.

NCW involves a number of interrelated concepts that form an intellectual basis for DoD transformation. "NCW is about human and organizational behavior. NCW is based on adopting a new way of thinking—network-centric thinking—and applying it to military operations. NCW focuses on the power that can be generated from the effective linking or networking of the enterprise."[2]

---

[1] Network Centric Warfare – A DoD Report to the Congress 27 July 2001. Executive Summary, pi.
[2] Alberts, Garstka, and Stein. *Network Centric Warfare: Developing and Leverage Information Superiority*. Washington, DC: CCRP Publications Series. 1999. p88. Originally, this quote contained the phrases *combat power* and *warfighting enterprise* instead of military operations. I made the change that, in effect, extends the NCW quote to NCO so that it is more reflective of the full mission spectrum.

NCW involves actions and their effects in four domains: physical, information, cognitive, and social. The fundamental capabilities that characterize a network-centric enterprise, organized by domain, are as follows:[3]

- Physical Domain: All enterprise entities[4] are robustly networked, achieving secure and seamless connectivity and interoperability.

- Information Domain: All participants have the capability to share, access, and protect information, not only within their organizations, but with other appropriate enterprise entities as well as others. Participants have the capability to collaborate in the information domain and to individually or collectively conduct information operations.

- Cognitive / Social Domains: Each participant has the capability to develop high quality awareness and to share this awareness. The enterprise has the capability to develop shared awareness and understanding, including an understanding of command intent. Participants are capable of self-synchronization.

Tenets of Network Centric Operations

NCW requires[5] the existence of a robustly networked enterprise (networked not just in the information domain, but also in the social domain as well). The value chain of a network-centric enterprise is as follows:

- Robustly networking an enterprise leads to widespread information sharing and collaboration.

- Increased sharing and collaboration improve both individual and shared awareness.

- Shared awareness improves decisions and, in the presence of edge approaches to command and control, enables self-synchronization.

- The result is dramatic improvements in mission / enterprise effectiveness and agility.

Integral to transforming network-centric concepts into fielded capabilities is the co-evolution of mission capability packages[6] and an understanding of the network-centric maturity model.[7]

Mission Capability Packages and Coevolution

The concept of a coevolved mission capability package (MCP) dates back about a decade.[8] Basically, coevolved MCPs are a response to the problems that can arise when new technology,

---

[3] Adapted from: Alberts, Garstka, Hayes, and Signori. *Understanding Information Age Warfare*. Washington, DC: CCRP Publications Series. 2001. p57.

[4] The word *enterprise* is used here instead of *force* in recognition of the requirement to conduct a wide range of coalition operations.

[5] This is not an all-or-nothing requirement. Improvements in networking will lead to commensurate improvements in information sharing and collaboration.

[6] First proposed in INSS Strategic Forum of the same name.

[7] Alberts et al., *Understanding*. Figure 76, p241.

other capabilities, or new concepts are introduced but are not accompanied by changes in other areas. For years, many people (some very knowledgeable) subscribed to the popular view that computer technology was not cost-effective because there was a lack of empirical evidence to show improvements in productivity. In fact, there was more than a grain of truth in this view, but it is more appropriate to conclude that a lack of coevolution was the culprit, not a lack of the potential power of computers. To make matters even worse, a lack of coevolution or co-adaptation cannot only prevent value from being created, but can also have significant adverse impacts. Thus, the introduction of computers or new ideas that have enormous potential can actually degrade performance.[9]

With respect to changing the current approach to planning to one that is better suited to Network Centric Operations, these new network-centric mission capability packages need to include new approaches to command and control that specify the nature of the planning process, the plan it produces, the systems that support C2 (specifically planning), organization including roles and responsibilities, interactions among participants, the distribution of information, recruitment objectives, education and training programs, and personnel practices and incentives. The approach to execution would also need to be considered. A coevolved MCP is one with elements that have all been adapted to be mutually supportive of the concept of operations.

Network-Centric Maturity Model

The network-centric maturity model, depicted in Figure 1 below, defines 5 levels of maturity and a hypothesized migration path for the implementation of network-centric capabilities in an organization.

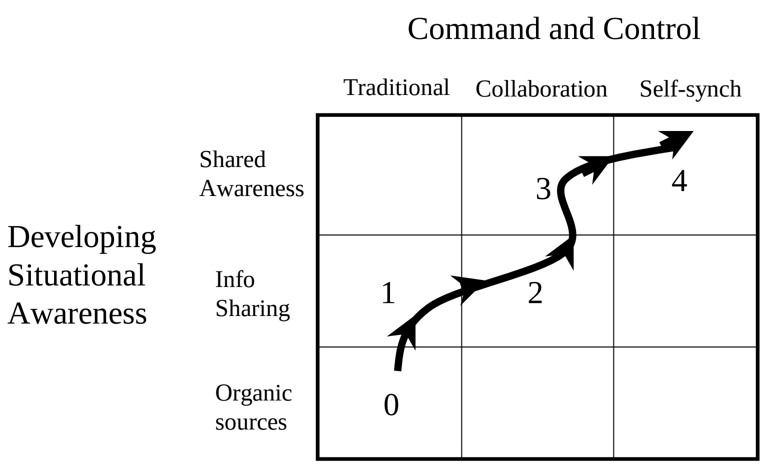


Figure 1: The Network-Centric Maturity Model

In a pure platform-centric, stove-piped world, sensors are owned by platforms and the information available to those on a platform comes, for all intents and purposes, solely from these sensors. Thus, situation awareness is developed only from organic sources. Level 0, the

---

[8] Alberts. *Mission Capability Packages*. INSS Strategic Forum 14. Washington, DC: NDU Press. 1995. Alberts. *The Unintended Consequences of Information Age Technologies*. Washington, DC: CCRP Publications Series. 1996. pp47-52.

[9] Alberts, *The Unintended Consequences*. pp13-14.

baseline, is defined as operations that employ traditional command and control processes (e.g., centralized planning) with an information position that is created solely from organic sources. Level 1 is a mini-step in the direction of Network Centric Operations. Although it employs traditional approaches to command and control, it involves a significant amount of information sharing among the participants in an operation. Level 1 requires a somewhat connected (networked) force, but without a requirement for information services beyond that necessary to support information sharing.

Level 2 takes the next step and introduces some collaboration among participants across location, function, and organization. The collaboration involved here is focused on the nature of the information being shared in order to identify inconsistencies and sort out incorrect, out of date, or questionable information. Level 2 requires a more robustly connected network with a collaboration environment.

Level 3 differs from Level 2 in the nature of the collaboration and the level of shared awareness that is achieved. At Level 3, collaboration is focused not on the information but on what it means, its implications, and the nature of a response. Thus, the collaboration that takes place at Level 3 involves an increase in distributed decision rights.

Level 4 takes the final step to Network Centric Operations and involves an approach to command and control that at least permits, if not encourages, self-synchronization. Self-synchronization requires a significant increase in the distribution of decision rights. Network-centric enterprises are not created by effecting incremental improvements, but by requiring disruptive innovations. This involves coevolution, that is, changes to many, if not all, of the elements of a mission capability package—a significant departure from business as usual. Changes to the way the functions of command and control are accomplished are essential to network-centric mission capability packages. The maturity model posits stages in the transformation of an organization that may be used as a basis for an evolutionary approach that can be employed to manage the risks associated with a transformation to a network-centric enterprise.

Effects-Based Operations

NCW/O and EBO are complimentary concepts. In fact, NCW is the starting point for EBO.[10] While NCW/O embodies a set of principles (more recently elaborated upon in the articulation of Power to the Edge[11] principles, policies, and practices) that address how an enterprise functions, EBO speaks to the nature of command intent and to the need to focus on the ultimate effects that are desired rather than solely upon the actions taken or the immediate effects of these actions. The fulcrum of NCW is shared awareness. Efforts can be focused on getting to shared awareness and being able to leverage shared awareness. Awareness includes an understanding of intent, where the quality of the intent will be a function of the degree to which both the immediate and consequential effects (a cascade of effects) in all of the relevant domains are understood.

[10] Smith. *Effects-Based Operations*. Washington, DC: CCRP Publication Series. 2002. p59.
[11] Alberts and Hayes. *Power to the Edge*. Washington, DC: CCRP Publication Series. 2003.

EBO recognizes that among the most important effects are those that are perceived by various entities because they affect both will and ability. EBO places military or civil-military operations in context, seeing the role of the military as establishing the environment (secure and peaceful) necessary to achieve political, economic, and social objectives.[12]

Command and Control

New approaches to both command and control are necessitated by (1) a need to accommodate the realities of coalition and civil-military operations and (2) a desire to increase awareness and leverage shared awareness. Command needs to be separated from control. Command should be equated with the establishment of a set of initial conditions, including how these conditions are adjusted dynamically. Control needs to be looked at as an emergent property, one that is a function of initial conditions including those that are established by command.

Command is a scalable concept. It applies at the enterprise level where it might be referred to as governance and it applies at the small unit or individual level where a specific task or mission is involved. At each level, understanding command begins with an understanding of what the functions of command are. Command functions include:[13]

- Establishing the goal or objective (the intent)
- Determining roles, responsibilities, and relationships
- Establishing rules and constraints
- Monitoring and assessing the situation and progress

There are a set of other functions that are traditionally associated with commanders that include inspiration, motivation, training, and preparedness. These are the functions that are often associated with leadership. For the purposes of this examination of network-centric planning in the context of the larger issue of network-centric command, these leadership functions will not be considered.

There are many different approaches that have been taken to accomplishing the functions associated with command that have proven successful. Traditional military approaches are, however, a reflection of Industrial Age thinking and capabilities. Six Industrial Age approaches or philosophies were successfully employed by 20th century militaries.[14] These approaches "decomposed the battlespace *(or problem)*, created layered organizations, divided into specializations, and organized forces into hierarchies."[15] The assumption that was made was that this approach to C2 and organization was what was needed to transform the complexity of war and large operations into a collection of simple, manageable tasks—tasks that, if accomplished individually, would collectively accomplish the larger mission. A lot of time and effort was spent in finding ways to (1) optimize the performance of individual tasks and (2) deconflict the units that were undertaking these tasks from adversely impacting one another. Deconfliction efforts involved several dimensions: functional, spatial (geographical), and temporal. Only complicated

[12] Smith. *Effects-Based Operations*.
[13] This material is taken from a forthcoming DoD/CCRP book entitled *Understanding Command and Control,* expected to be published later in 2005.
[14] For a discussion of Industrial Age approaches to Command and Control see: Alberts and Hayes, *Power to the Edge*. pp18-26.
[15] Alberts and Hayes, *Power to the Edge*. p44.

problems, that is, problems that, regardless of the number of parts, behave in a linear fashion and whose behaviors can be adequately predicted, are amendable to this Industrial Age approach to C2.

The objective of command is to create the best conditions possible, conditions that are more, rather than less, likely to give rise to the desired result(s), or the converse, less likely to have less desirable results. Thus, the functions associated with command involve understanding the risks inherent to the situation and managing those risks.

Sensemaking

Command is about accomplishing a set of functions. The successful accomplishment of those functions requires an individual, team, organization, or collective to be able to make sense of the situation. Making sense of the situation, *Sensemaking*, begins with putting available information into context and identifying the relevant patterns that exist.[16] In other words, sensemaking begins with the development of situation awareness. Situation awareness[17] includes awareness of intent (purpose, considerations, and constraints). The intent of interest is not only of the mission at hand and the participants in this mission, but also an adversary's intent. In the process of developing situation awareness it may be determined that more information is needed before a response can be formulated. As a result, a decision (command) may be made to task collection and analysis assets. Sensemaking involves more than developing situation awareness, it goes beyond what is happening to what may happen and what can be done about it. This involves prediction and analysis, both of which require a model (mental or explicit) and the knowledge of or development of options that map to various alternative futures.

The need to consider a wide range of effects and the cascades of effects that take place in multiple domains and contexts (social, economic, political) requires more knowledge, experience, and expertise than when the only effects that were considered were direct military effects. This is one of the reasons that effects-based planning benefits from a network-centric approach.

At some point, decisions regarding what to do about the situation, in the form of a proactive strategy to shape events or in the form of a response to events, are made. Responses include not only direct action(s) but, as a result of a conclusion/decision that more information is needed, action(s) to get additional information, either directly by tasking some other entity, or by initiating an information exchange. As far as a determination of whether to be proactive or reactive, command decisions could involve taking actions or taking steps to prepare for taking action (again these decisions could be to delegate, form a team, or do it). Preparing for action is called *planning*.

Mission Planning

Planning is one of the activities included in a command and control process. Because the command function determines intent and establishes a set of initial conditions for, at the

[16] Alberts and Hayes, *Power to the Edge*. p101.
[17] Alberts et al., *Understanding*. pp120-125.

enterprise level, the entire organization and, at the mission or task level, for the specific undertaking at hand, it shapes both the substance of and the processes of planning.
Mission planning[18] involves preparing for a response. Mission planning begins with an understanding of command intent in the context of the specific situation. The planning process involves an interpretation or amplification of intent and its re-expression in the form of a plan. Thus, planning processes serve to accomplish a number of the functions associated with command and control. Planning includes both decisionmaking and anticipatory decisionmaking (contingencies). A plan should address the following elements:[19]

- An (re)expression of intent
- Allocation of roles and responsibilities
- Allocation of non-organic assets
- Setting boundaries and establishing constraints
- Establishing milestones and schedules
- Articulation of contingencies

Decisions regarding the allocation of resources, the setting of boundaries, etc. are subject to inherited[20] intent and established conditions and constraints (if any).
The nature of the process that produces the set of decisions that are reflected in a plan and the nature of the plan itself (its expression and level of detail) can vary significantly as a function of the operable approach to command and control. Plans can be explicit or implicit. They can vary in the degree of detail/granularity from just an expression of intent to a complete specification of the "who, what, where, when, and how." The process of planning can be centralized or decentralized, formal or informal, exclusive or inclusive, cyclical or dynamic and interactive, authoritative or collaborative.

A hallmark of traditional approaches to command and control is centralized planning.[21] Given the limits of Industrial Age communication (and information gathering and dissemination), plans were the mechanisms by which military commanders sought to impose their will on their organizations as well as on the situation. Large organizations dealing with major operations created comprehensive and fairly detailed plans. These plans required considerable time and resources to develop and, because of the dynamics of the situation, needed to be adjusted. A classic large-scale, Industrial Age plan is the Air Tasking Order (ATO), perfected during the latter decades of the 20th century.[22] The ATO can be thought of as a specification of a *course of action*, a set of actions to be taken by various entities, arranged both temporally and spatially. Despite automation and process improvements, the ATO process requires a relatively large headquarters with highly skilled and trained individuals and a considerable amount of time (72

[18] Mission planning is the focus of the campaign of experimentation discussed here. Planning, like command and control, can occur at any and all levels and serves to support the functions of C2.
[19] Adapted from: Alberts and Hayes, *Power to the Edge*. p47.
[20] *Inherited* here refers to a chain of command. The chain, while formally established in military and other hierarchal organizations, need not be formal nor static.
[21] This discussion of centralized planning is paraphrased from: Alberts and Hayes, *Power to the Edge*. pp46-50.
[22] Air Tasking Order: A method used to task and disseminate to components, subordinate units, and command and control agencies projected sorties, capabilities and/or forces to targets and specific missions. Normally provides specific instructions to include call signs, targets, controlling agencies, etc., as well as general instructions. Also called ATO. (Department of Defense Dictionary of Military and Associated Terms. Joint Publication 1-02. http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf)

hours) to produce an ATO that tasks every aircraft and provides the information they need. The emphasis of the process is on the deconfliction of airspace.

While this approach may work well enough in a somewhat static environment of fixed targets, it is less than ideal in a more dynamic situation with changing priorities and moving forces and targets, and even less effective for irregularly dispersed forces and fleeting targets. Plans for air operations can have an adverse effect on ground operations, constraining them by placing certain areas off limits to friendly ground forces in order to prevent fratricide. Thus, the effectiveness of air and space planning needs to be viewed in the context of the overall operation, taking into consideration the needs of the joint or coalition force and the totality of the effects that are sought.

Air and Space Operations: C2 Challenges

Major differences exist between what can be characterized as Industrial Age air and space operations and those of the 21st century. These differences make the latter far more complex, more time sensitive, and more difficult to assess.

The increased complexity of air and space operations is a result of the increasingly interdependent and nonlinear nature of 21st century joint and coalition operations. In these 21st century operations, air and space power must be integrated with other force elements as well as other instruments of power. Air power can no longer operate in a standalone manner. The Industrial Age approach that involved a carefully sequenced air campaign that began with suppression of enemy air defenses and the achievement of air superiority as a prerequisite for subsequent air, land, and maritime operations has been replaced by a more simultaneous approach that requires greater sharing of information, awareness of blue forces and other entities, and collaboration with other force components as well as a variety of other actors. Time, always of great importance, is critical in light of rapidly changing situations, increasing lethality, asymmetric adversaries and tactics, and a 24-hour news cycle. The challenges of battle damage assessment, exacerbated by improved techniques at cover, concealment, and deception, still pale in comparison to those posed by effect-based assessments. The ability to effectively deal with increased complexity, a faster tempo, and operations that are more difficult to assess, while a significant challenge in and of itself, is made more challenging with increased strategic, operational, and tactical uncertainty.

The appropriate response to these strategic and mission challenges is *agility*, both force and C2 agility. This makes the creation of an agile plan a primary consideration.
Agility[23] is a complex, multi-dimensional concept, a capability that has been associated with Command…Control…in the Information Age.[24] The dimensions of agility[25] are:

- Robustness
- Resilience
- Responsiveness

[23] Alberts and Hayes, *Power to the Edge*. pp123-59.
[24] Alberts and Hayes, *Power to the Edge*. pp201-212.
[25] Alberts and Hayes, *Power to the Edge*. pp127-128.

- Flexibility
- Innovation
- Adaptation

Agility, of course, has no value without the ability to be effective. Thus, for example, responsiveness does not mean the ability to react quickly, but the ability to react at an appropriate time and in an effective manner.

The need for a campaign of experimentation focused on developing a network-centric approach to air and space command and control, specifically the development and assessment of approaches to planning (and execution) in a network-centric environment, embodies the assertion that a network-centric approach to command and control is an approach that is more agile (and effective) then traditional approaches. This is not to say that network-centric approaches will always be preferred but that, in a significant and interesting part of the mission space where agility is a decided virtue, a network-centric approach to air and space command and control makes sense.

Campaigns of Experimentation

The Industrial Age approach of centralized planning and decentralized execution is in the process of being replaced by Information Age, network-centric approaches designed to seamlessly integrate planning and execution. While new patterns of information sharing and altered distributions of roles and responsibilities will be required, there are many questions that need to be answered and a wide variety of approaches that need to be explored and assessed before we will adequately understand when and under what circumstances these new approaches should be employed. To generate the information necessary to answer the questions (and ask the right questions) and to sort out the more promising approaches from the less promising approaches, a campaign of experimentation, accompanied by focused research, needs to be undertaken.

A Campaign of Experimentation is not just a collection of experiments but "a process that (1) combines and structures experimental results much in the way that individual bricks are fashioned into a structure for a purpose, and (2) steers future experimentation activities."[26]

Nature of a Campaign of Experimentation

A campaign of experimentation is a journey from the general to the specific. In this case, it involves moving from concepts to capabilities and from hypotheses to understandings. A campaign of experimentation is a balancing act involving tradeoffs between variety and replication. Unlike the natural progress of science where experiments are, at the very best, loosely coupled and depend on the abilities and desires of a set of independent actors, a campaign of experimentation is a focused activity where the selection and sequencing of experiments and research activities are a reflection of coherent decisionmaking. If properly

[26] Alberts and Hayes. *Campaigns of Experimentation*. Washington, DC: CCRP Publication Series. 2005. p2.

conceived, designed, and executed, campaigns of experimentation can be far more efficient and achieve results in a shorter period of time.

To the extent that they are orchestrated and hence reduce choice, campaigns of experimentation are subject to a number of risks.

Risks and Remedies

Joint Forces Command (JFCOM), each of the Services, and DoD Agencies are, to different degrees, experimenting with new ways to accomplish their assigned missions and responsibilities. These activities have, for a variety of reasons, not been as productive as they could be. The following five reasons have been identified for the observed shortcomings:[27]

1. Moving ahead without sufficient evidence and understanding.
2. Prematurely settling on an approach.
3. Confining explorations to the Industrial Age–Information Age border.
4. Failing to capitalize on the creativity present in the force.

The campaign of experimentation and accompanying research program outlined here is designed to avoid these known pitfalls. How this is accomplished will be discussed later.
Focus and Objectives

Campaigns of Experimentation are all managed to some degree. The nature of the management structure and approach can vary, just as there are different approaches to command and control. The articulation of a campaign's focus is analogous to the articulation of command intent.
In this case, the focus is on exploring and assessing network-centric approaches to effects-based planning of air and space operations that are part of NCO.

Phases of a Campaign of Experimentation

The journey from the general to the specific, moving from concepts to hypotheses to understanding and ultimately to, in this case, a network-centric planning capability, needs to proceed in an orderly fashion, paced not according to a predetermined schedule but rather paced by actual progress, by the achievement of specific milestones. Campaigns begin, according to the Code of Best Practice,[28] with a Formulation Phase, followed by a Concept Definition Phase and a Refinement Phase, and conclude with a Demonstration Phase.

Formulation Phase

All campaigns begin with an idea, in this case the idea of network centric planning as an integral part of a network centric approach to command and control. The first phase of a campaign, Formulation, takes the idea for the campaign, one that is usually expressed only in general terms,

---

[27] Alberts and Hayes. *Campaigns of Experimentation*. pp21-28.
[28] For a discussion of the phasing of a campaign of experimentation see: Alberts and Hayes. *Campaigns of Experimentation*. pp117-124.

and develops a statement of goals and objectives. It sketches out the landscape to be explored and understood and the capability to be developed. The Formulation Phase also involves decisions regarding an initial allocation of resources, the participants in the campaign, and their respective roles and contributions. The products of this phase include a statement of goals and objectives expressed in the context of an initial version of a conceptual model.

Concept Definition Phase

With the general statement of goals and objectives as a point of departure, the Concept Definition Phase is focused on, as its name implies, developing more precise definitions, the construction of a working conceptual model, the construction of the treatment space, and the development of metrics that define their attributes and instruments used to take measurements of these attributes. In other words, the Concept Definition Phase makes the idea or concept operational and begins to explore the landscape.

This phase consists of the undertaking of discovery experiments and the conduct of exploratory analysis. The products of this phase are (1) an enriched conceptual model, (2) a set of testable hypotheses organized by priority and sequenced, and (3) measurement tools and instruments.

Refinement Phase

The Refinement Phase is where indepth undertakings are developed and capabilities built. Guided by the state of the conceptual model developed during the Concept Definition Phase, the activities undertaken during this phase test hypotheses and refine and assess capabilities. While during the Concept Definition Phase there is a need to make sure that all of the possibilities are considered, in the Refinement Phase there is a need to make sure that these early successes are adequately retested and that the conditions that are needed to obtain success are well understood. This involves pushing ideas and capabilities to their breaking point, developing a solid understanding of the relationships among key variables and developing an understanding of why various approaches or concepts (treatments in the language of experimentation) work and do not work.

This phase usually takes the longest and requires the most resources. It is also the phase that, if properly conceived, will have its share of unexpected setbacks and pleasant surprises. The products of this phase include (1) a refined and enriched conceptual model and (2) capabilities that are ready to demonstrate.

Demonstration Phase

Demonstration experiments and confirmatory analyses form the core of the Demonstration Phase. The purpose of this phase is to provide an opportunity for a variety of individuals and organizations to better understand the new concepts and experience the new capabilities. This phase presents the evidence for adoption of the new concepts and capabilities.

It is important that this phase does not begin until the concepts are adequately explored and the capabilities are adequately tested in the previous phases. Prematurely moving into demonstrations is a common, avoidable mistake.

Conducting a Sound Campaign

Conducting a successful campaign of experimentation is not easy. In addition to the obvious, that it requires considerable knowledge, expertise, and experience, it also requires a great deal of patience and a willingness to accept ideas that run counter to conventional wisdom. It also requires sound management and fiscal flexibility.

Among the most critical items that are needed to avoid problems and fully leverage the opportunities that present themselves are:

- Conducting peer reviews
- Employing objective measures of effectiveness or value
- Documenting experiments and analyses
- Keeping the conceptual model current
- Conducting analysis beyond individual experiments
- Avoiding a premature narrowing of the focus
- Maintaining continuity of personnel

These are discussed in the applicable codes of best practices (COBP) and elsewhere, specifically the NATO COBP for C2 Assessment, the COBP for Experimentation, and the COBP for Campaigns of Experimentation. All can be found at www.dodccrp.org.

Expectations must also be realistic so as to avoid creating situations that will compromise individual experiments and analyses, or worse, the campaign itself. Unrealistic expectations compounded by inadequate resources and an inflexible schedule have frequently resulted in flawed experiments and analyses and a lack of opportunity to collect useful data or to leverage what useful data is collected.

Perhaps the most important thing to realize is that one experiment or analysis cannot be expected to do too much. Collecting a number of small experiments and creating a large event adversely affects the quality of these experiments because priorities become confused and there are usually conflicting objectives. Conducting a number of smaller events is far more preferable.

It needs to be understood that campaign progress will be nonlinear and that “failure,” as defined by a concept that does not work as expected, is not failure but a success as long as the experiment or analysis was done properly and is well documented. Hence, resourcing must be flexible to allow for additional experiments and/or analyses as required.

Each of the COBPs has a lessons learned section. Prior to undertaking this campaign of experimentation, each participant should review this material to help ensure that these avoidable problems are not repeated. Prior to the conduct of each individual experiment and analysis, the team should once again review the applicable lessons learned.

Conceptual Model

The *Code of Best Practice for Campaigns of Experimentation*[29] identifies two pre-conditions for success: (1) building a strong team and (2) creating an explicit conceptual model. A conceptual model is the heart of any experimentation campaign. A conceptual model is an explicit statement that organizes and synthesizes existing knowledge and guides the course of the experimentation campaign. Specifically, a conceptual model identifies (1) the set of relevant variables, (2) the set of relationships among them, and (3) specifies the measures of merit (MoM). The conceptual model will evolve over the course of any campaign of experimentation as experimental or analytical findings become available or known, whether from activities that are part of the campaign or the efforts of others. Thus, keeping abreast of related activities is important. When one selects a subset of the variables and/or relationships from the conceptual model, this is called a "view." Two specific views are central to a campaign of experimentation. The first is a "value view" that focuses on variables that are a reflection of quality or value and the relationships among these variables. The value view provides a value chain that can be explored, tested, and used during the campaign to design experiments or analyses. The second view of interest is the "process view." A process view looks at a set of variables that describe various processes including inputs and outputs. A process view is used to provide data that informs a value view. For example, the process view provides the transfer function between inputs (e.g., information) and outputs (e.g., awareness). The characteristics of both the inputs and the outputs are reflected in measures of quality (e.g., information quality and quality of awareness). Thus, the process view, when instantiated for a particular case, provides one data point for the relationship between two measures of value. When process-related experiments or analyses are repeated over a range of conditions (the variables that moderate this relationship), an estimate of the nature of the relationship between the value variables can be developed.

Conceptual Model for Network-Centric Planning: Overview

As indicated previously, a conceptual model provides the framework that is needed to organize what is known and what is not known, as well as guide both experiments and analyses. In this section, an initial conceptual model of planning in the context of NCO is developed. The model provides a point of departure for the campaign and will be enhanced and refined during the course of the campaign.

The level of detail and specificity of this model is consistent with what would normally be generated during a campaign's formulation phase. This model contains variables that reflect the key concepts and the relationships among them including a set of variables that form a value chain that provides a basis for comparing the existing approach to ATO planning to network-centric approaches.

This section provides a high-level overview of a conceptual model that can be employed in a campaign of experimentation to explore network-centric approaches to planning.

[29] Alberts and Hayes. *Campaigns of Experimentation*. pp100-101.

Figure 2 is a high level depiction of planning in the context of command and control and operations. It depicts the relationships among Command, Control, Sensemaking, Planning, and Execution.

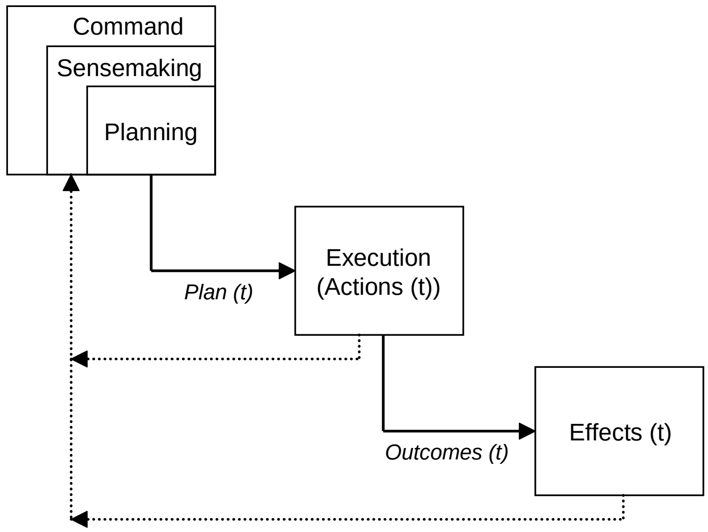


Figure 2: Planning in the Context of C2 and Operations

Command initially, and control during the course of an operation (as long as the situation remains within the parameters set by command), determines the conditions that shape the sensemaking process. These functions shape both the substance and the nature of the sensemaking process. They shape the nature by determining goals and objectives as well as the value proposition that determines what is desirable and what is not. They shape the substance of the sensemaking process by determining roles and responsibilities, allocating resources, and determining the nature of the interactions that take place among the participants in the sensemaking process.

Planning is shown as an integral part of the sensemaking process. Planning is about the visualization of alternative futures and once a course of action is selected (a function of command), planning refocuses on preparing for action. The product of planning is a “plan,” an expression of the set of decisions that defines the selected response to the situation as well as contingencies mapped to future conditions. Plans can vary greatly in the level of detail they contain.

The plan is then executed. Execution consists of a set of actions taken in various domains, and distributed in space and time. These result in changes to a set of state variables and in turn create a set of effects in various domains, distributed in time and space. To the extent that specific actions are specified in a plan, they are usually linked to one another either directly or conditionally as a function of the situation and/or the effect(s) that they were intended to create.

The Functions of Command

The functions of command, sensemaking, and planning (see Figure 2) are accomplished concurrently and interactively. Command establishes important conditions that affect the sensemaking and planning processes. Sensemaking determine what needs to be accomplished and to some degree of specificity, the approach. Planning fills in the gap from an expression of intent to actionable decisions that can be understood and executed.
Figure 3, Critical Command Functions, identifies the key command functions, putting them in the context of the domains.

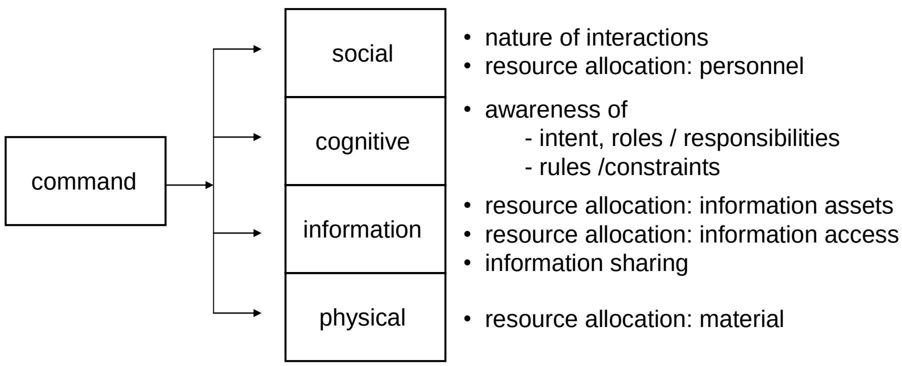


Figure 3: Critical Command Functions

In the social domain, command establishes the rules that govern interactions among participants and participating organizations and allocates resources. Another function of command is to articulate command intent, assign roles and responsibilities, and establish rules of engagement and constraints. The actions of command that convey the decisions result in a particular state of awareness regarding command intent and direction that exists in the cognitive domain. Among the rules that govern interactions and the resources that are allocated are those regulations of the information domain, specifically those that affect the use of information assets, access to information, and information sharing. These are critical enablers of network-centric approaches and need to be considered controllable independent variables that, in part, determine the approach to C2/planning. The allocations of materiel are located in the physical domain.

Sensemaking

Figure 4, Sensemaking, identifies key concepts related to sensemaking and the relationships among them. This figure also puts these concepts and relationships in the context of the social, cognitive, and information domains.

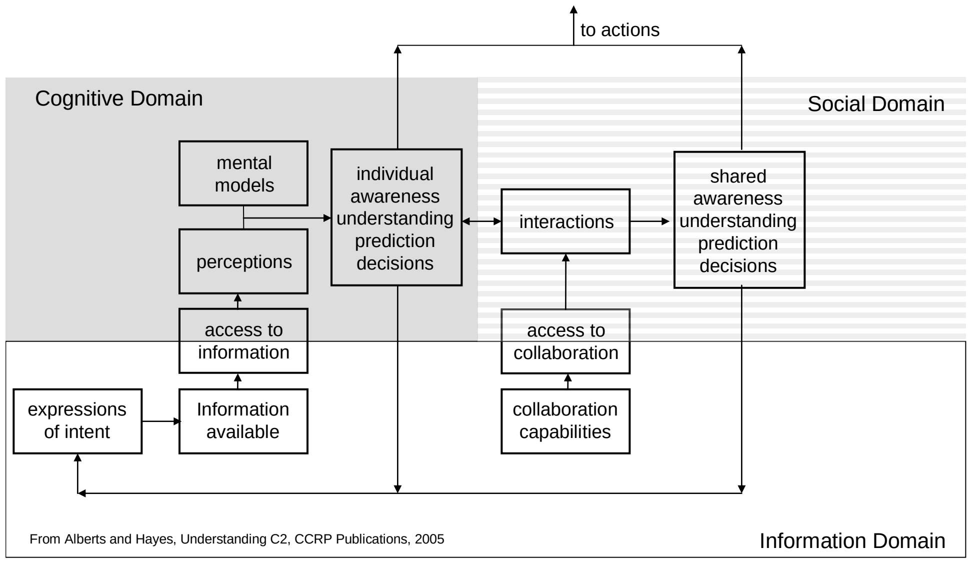


Figure 4: Sensemaking

The key outputs of a sensemaking process that involve two or more entities are individual and shared awareness, understanding, prediction, and decisions. The individual and shared decisions made lead to actions.

Planning

Planning is an integral part of sensemaking and it is not productive to attempt to definitively define where the other components of sensemaking leave off and planning begins. Like command and control, sensemaking and its planning component involve decision making and are scalable. Sensemaking can be undertaken by a single individual, a couple, a small team, and by large collections of individuals that are members of an organization. Indeed planning may be undertaken by groups of individuals from various organizations. As a process, planning involves the identification and sequencing of the tasks required to achieve the command intent. These tasks may be undertaken by individuals, groups of individuals, and organizations interacting to varying degrees with one another.

Effects-based planning involves the identification and sequencing of desired effects as an input to the process that identifies and sequences the necessary tasks.

Dynamic Planning refers to a planning process that interacts in near-real time with execution as opposed to traditional planning where planning and execution are cyclical.

Planning and Execution

Traditionally, planning is thought of as a separate and distinct process and activity from execution. Traditional notions of planning and execution assume an Industrial Age organizational structure with planning occurring at the headquarters (operational) level and execution involves subordinate units. However, one ignores at their peril the fact that these subordinate units also engage in sensemaking (including the planning that is necessary to carry

out the plans that headquarters produce). Thus, traditional notions of planning and execution are distinguished from one another more by the nature of the distribution of decision rights than by the nature of the activity that is involved. Traditional notions of planning and execution implicitly assume a certain approach to C2 and organization.

It has been recognized that this separation between planning and execution inhibits the ability to rapidly respond to changing situations, the solutions currently under consideration involve, for the most part, incremental changes to process. For example, a "collaborative" approach to planning in which "executors" can listen in, and if necessary, interject.

The need to or the desirability to separate planning and execution (by function or by organization) should not be taken as a given. In fact, one of the subjects of a campaign of experimentation focused on planning should be the nature of the interactions between these two intimately related processes and the organizations (individuals) that are involved.
Figure 5 provides a generic process view of the traditional approach to planning in relationship to execution. This generic model, when instantiated for a particular process, can serve as a baseline in an analysis that explores non-traditional or network-centric approaches to planning and execution. Experiments that look at the relationship between planning and execution should be part of a larger campaign of experimentation devoted to finding and assessing new approaches to planning.

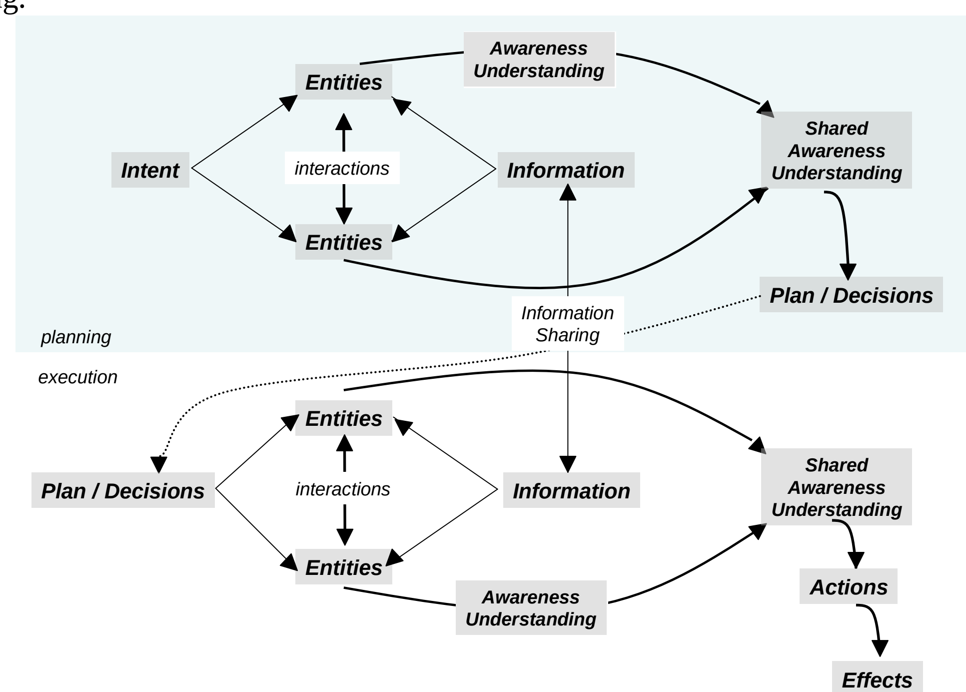


Figure 5: A Generic Process View of Traditional Planning and Execution

The sensemaking aspects of execution in the traditional approach to planning and execution, as depicted in Figure 5, constitute fractals that are related to the sensemaking that takes place during planning. Planning processes and activities are usually nested and differ from one another with respect to scale and the nature of delegated responsibilities and authorities. The process labeled

planning begins with an expression of (command) intent. The expression of intent can vary in its specificity. In turn, planning processes produce a set of decisions that collectively constitute what is called a *plan*. The nature of a plan can differ widely from an interpretation or re-expression of intent to a set of specifications about what actions are to be taken, when and where they should be taken, and even how they should be taken. Thus, the nature of the sensemaking involved in these processes differs as well. A major difference is the nature of the entities involved (e.g., planners, representatives of different functional capabilities, executors, or some combination). The processes depicted can also differ with respect to how and to what extent information is shared and the nature of the interactions among the participants. The interactions that take place determine, in part, at which network-centric maturity level the organization is operating.

Because this campaign is about both Network Centric Operations and effects-based operations, Figure 5 depicts not only the usual link between actions and effects but also a link between decisions and effects. This is because the decisions made, particularly when made known to an adversary, can have a profound effect on the cognitive state of the adversary independent of the actions taken.

Planning Space

In order to permit a full exploration of different approaches to planning, different planning processes, and different types of plans, it is important to define these, not as practitioners understand and practice them, but by (1) their functions and purposes and (2) the space of possibilities.

The generic planning process involves entities, interactions among entities, information, the distribution of information, and the sharing of information. Entities have assigned to them, or take on, roles and responsibilities. Furthermore the process begins with an expression of intent and produces a product, a plan, or set of decisions.

The Planning Space encompasses all of the variations that are possible across these dimensions. Figure 6, Planning Space, graphically depicts a space defined by the following three dimensions:

- The nature of the planning process
- The nature of the plan produced
- Information dissemination and sharing

The dimension “nature of the planning process” is anchored at one end by hierarchical processes commonly in use in military organizations. This would be typical of headquarters that focus on planning. At the other end of this dimension are edge approaches where planning, to the extent that planning is done, involves peer-to-peer interactions. The “nature of the plan produced” varies from a detailed plan, typified by the Air Tasking Order that provides an assignment to each wing number, to an expression of intent. “Information dissemination” can, on the one hand, be restricted to following the chain of command to being universally posted so that it can be pulled by those who need it.

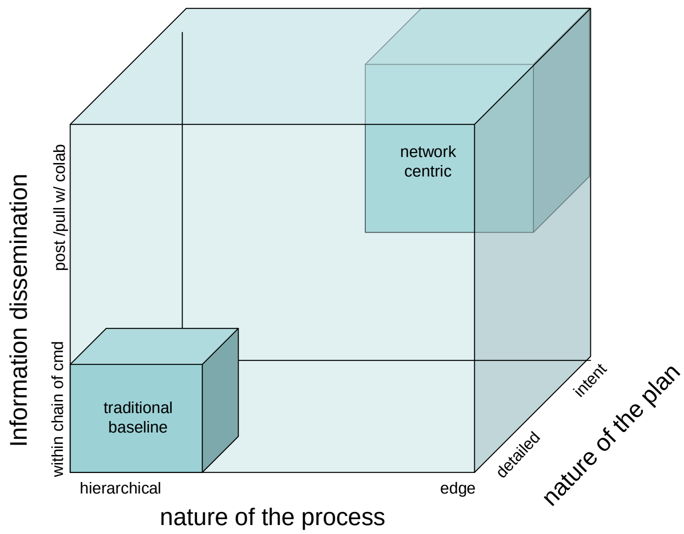


Figure 6: Planning Space

Traditional approaches to planning occupy the lower left of this space while network-centric approaches occupy the upper right.

The approaches to planning that need to be explored (actually, their locations in the Planning Space) are the primary independent variables for the campaign of experimentation. Traditional approaches also need to be analyzed and serve as the baseline for comparisons.

The Value View and Measures of Merit

Value-related variables or measures of merit are used to differentiate how well the functions depicted in Figure 5 are performed and the value or desirability (from someone's perspective) of the effects. Figure 7, Measures of Merit for Planning, depicts the value-related measures that are needed to explore different approaches to planning.

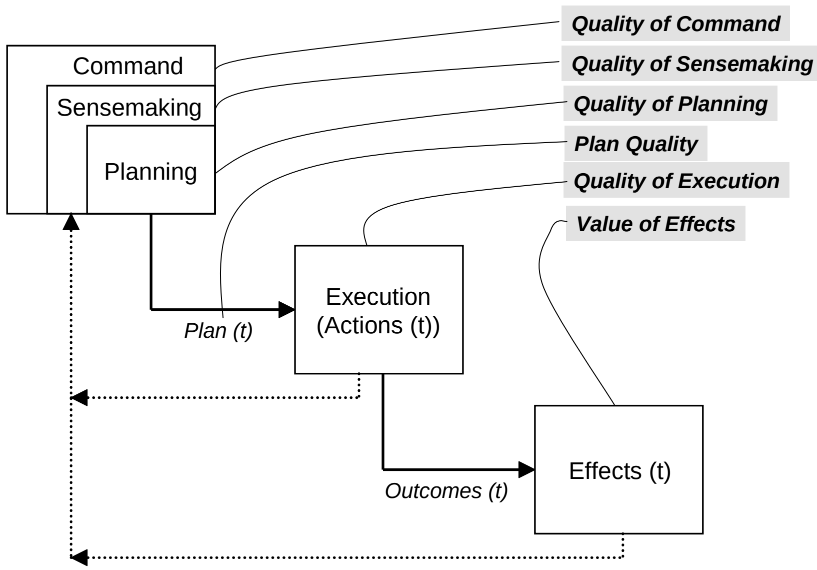


Figure 7: Measures of Merit for Planning

The value of outcomes was purposefully omitted from Figure 7 because the purpose of this campaign of experimentation is to explore planning for effects-based operations (effects-based planning). Exploring the hypotheses implied by Figure 7 to determine the extent of our knowledge will be a primary focus for the campaign in its early stages. For example, there is evidence that the quality of a plan makes a significant difference in mission outcomes and in the probability of achieving the desired effects. However, the ability of a plan to guarantee a good outcome is limited by a host of other factors including the quality of execution. Furthermore, the evidence that we do have is related to traditional approaches to C2 and planning.

This campaign of experimentation is specifically designed to explore new approaches to C2 and planning. We cannot simply assume that the relationships that have been previously observed between good planning and good outcomes will hold. In fact, the relationship between the quality of a plan and our ability to generate the effects we want may itself be a function of our approach to planning and our concept of what constitutes a plan.

For this reason we need to include in this campaign of experimentation experiments and analyses that test the hypothesized relationships among the measures of merit identified in Figure 7. The depiction of these relationships is referred to as the Value View and is found in Figure 8.

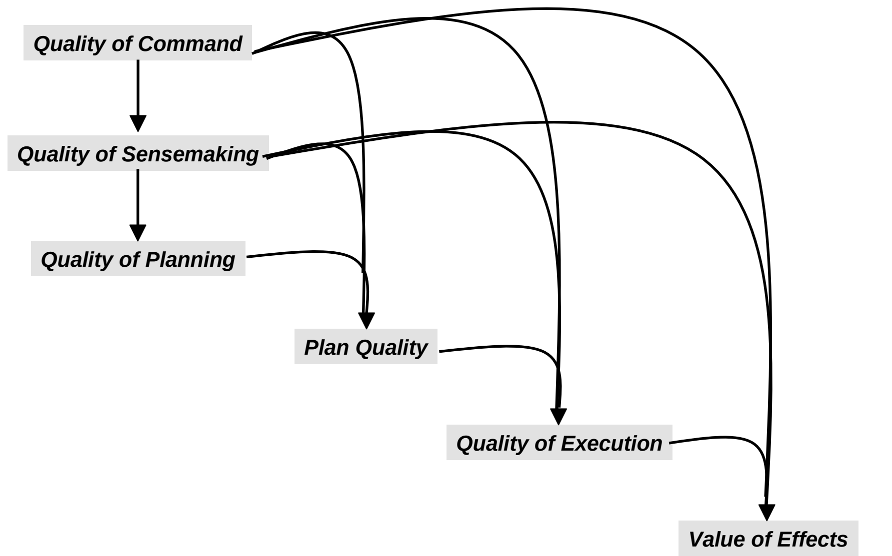


Figure 8: Conceptual Model: Value View

The Value View depicted in Figure 8 needs to be "operationalized" before it can be applied to specific analyses or experiments. This involves refining the quality and value concepts in Figure 8 to the point that we have put these concepts into the context of the network-centric value chain and decomposed these concepts into value-related variables that are amenable to measurement.
Quality of Command

Figure 9 provides the functions of command organized by domain.

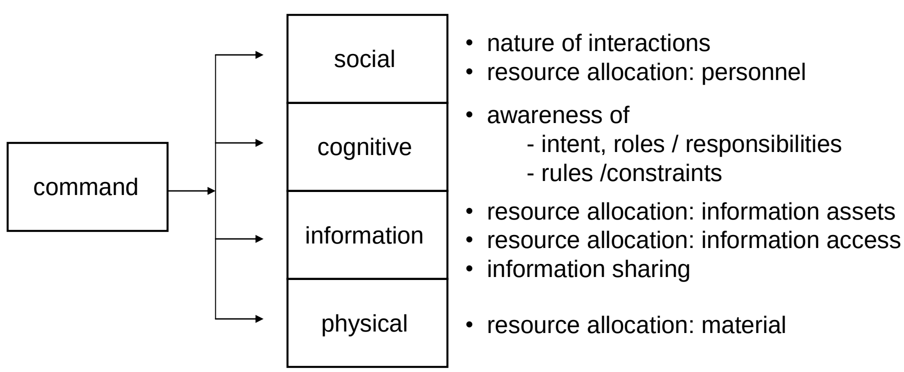


Figure 9: Functions of Command

The Quality of Command can be measured by how well each of the functions identified in Figure 9 is accomplished. For example, is the allocation of resources appropriate for the situation? Or, if there is an optimization model available, how does the allocation made compare with the optimal allocation?

Quality of Sensemaking

Figures 10, 11, and 12 relate the components of the Quality of Sensemaking to the tenets of NCW that form its value proposition. Figure 10 graphically depicts one of the major characteristics of a robustly networked force, namely access to information and collaboration that is provided across the force and specifically to participants in an operation.

A ***robustly networked force*** improves ***Information Sharing***

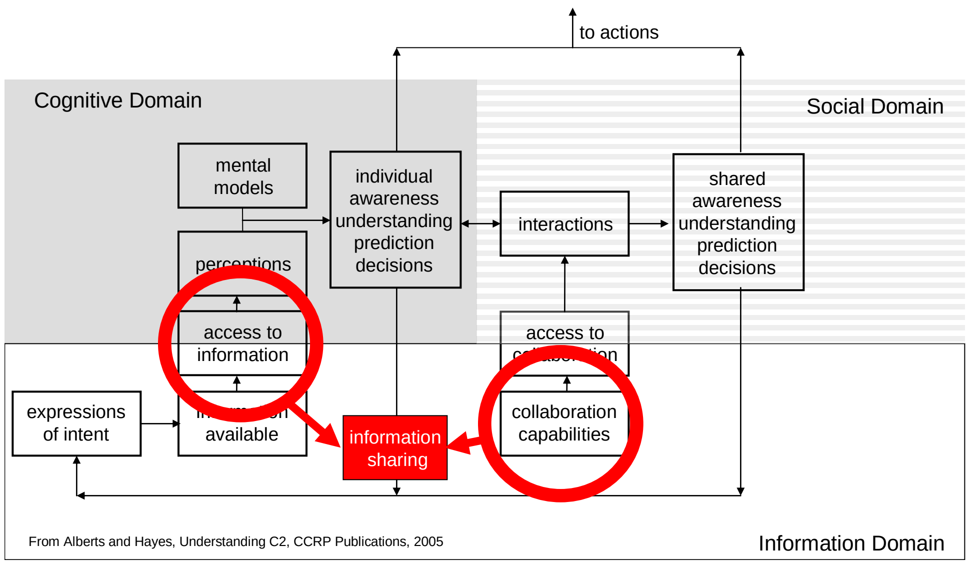


Figure 10: Value of Information and Collaboration

Figure 11 depicts the proposition that increased sharing of information (information domain) and collaboration (social domain) improves both the quality of information and the quality of shared awareness. Implicit in this statement is that the quality of individual awareness is also improved.

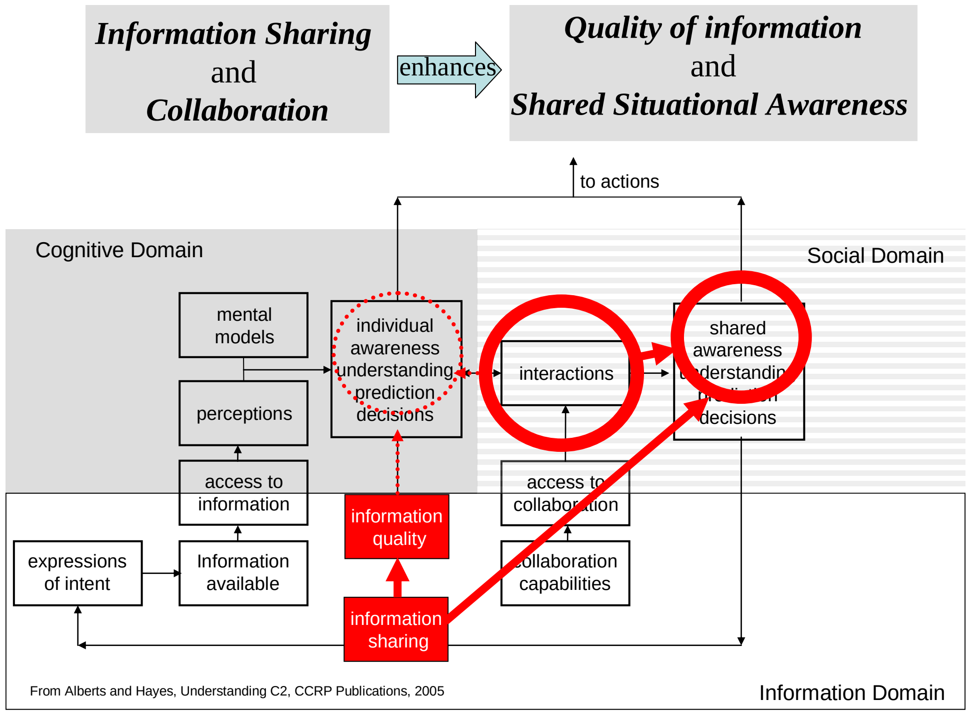


Figure 11: Improving the Quality of Information and Shared Awareness

The final link in the chain, depicted in the Figure 12, involves the relationship between shared awareness, collaboration, and self-synchronization. The hypothesis that, when the quality of shared awareness reaches some level, self-synchronization is enabled, is self evident. Nevertheless, this is one of the hypotheses that should be explored as part of this campaign. The apparent two-way relationship between collaboration and shared awareness may require some explanation. Collaboration is an umbrella term that applies to a variety of activities that vary in degree and intensity. Collaboration is a social domain activity but the object of collaboration can be in the information, cognitive, social, and physical domains. Sharing of information falls short of collaboration but is expected under the right circumstances to result in collaboration aimed at sorting good information from bad. This form of collaboration can evolve into types of collaboration that involve the meaning and interpretation of information. Collaborative decisionmaking usually involves a redistribution of decision rights. Self-synchronization is a form of collaboration in the action or physical domain. Thus, while collaboration in the information domain aimed at the sorting and understanding of information can lead to improved information, awareness, and shared awareness, it can also lead to richer forms of collaboration. The building of trust is an important factor in this movement up the scale of collaboration. The richer forms of collaboration, particularly those that involve a redistribution of decision rights, are enabled and facilitated by shared awareness. Thus, the two-way relationship between collaboration and shared awareness involves different types of collaboration.

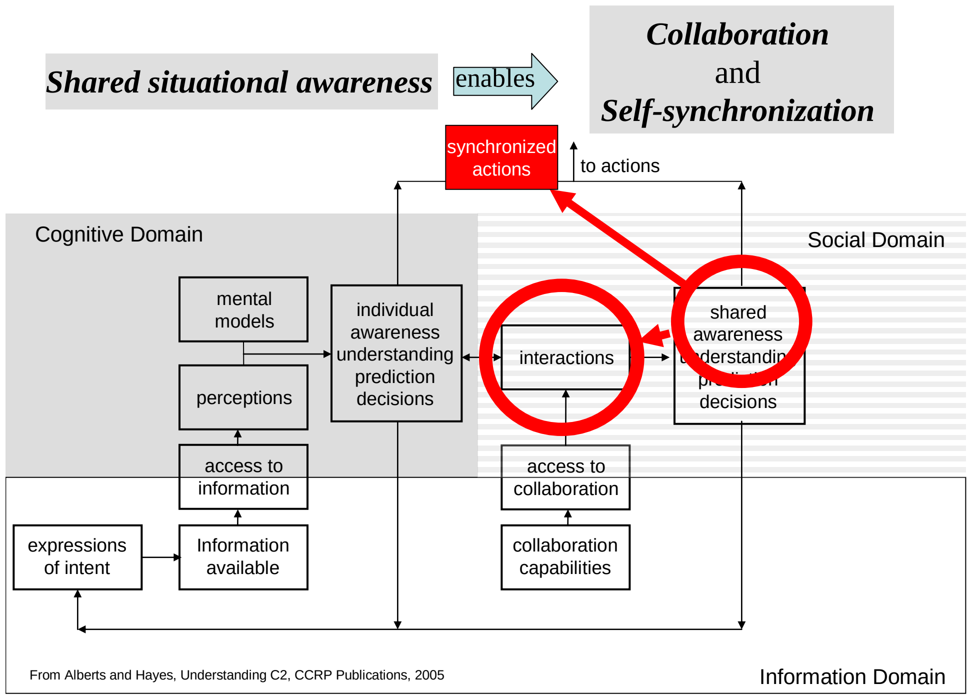


Figure 12: Achieving Self-synchronization

Quality of Planning and Plan Quality

The value in planning comes from the process and not the plan. This is because, as it has been said, "no plan survives first contact with the enemy."[30] As the situation changes and as plans fail to achieve their desired effects, plans needs to be reassessed and modified or sometimes even completed scraped. Traditional methods of assessing the quality of a plan looked at how long the plan remained viable, that is, how long did the plan survive without needing to be adjusted and how long did it survive without needing to be replaced?[31] This is not an adequate measure in and of itself and may, in fact, be misleading. This is because the length of time that a plan survives is a function not only of its quality but of how ambitious it is. More conservative plans may survive longer than aggressive plans, but in the final analysis may not be as effective in achieving the desired effects. In addition, different approaches to planning require different actions to be taken when things do not go according to plan, and thus what constitutes the survival of a plan differs. For these reasons, these traditional metrics and approaches to assessing the quality of a plan may not be particularly useful or may need to be supplemented with other measures.

There are both objective measures of planning process quality and plan quality (ones that can be measured relative to an objective standard) and fitness measures that relate to the quality of a plan in a specific context. Objective measures of the planning process include the time and resources required to produce a plan. An objective measure of plan quality would be its completeness (conditioned on the approach to planning that is being taken). An example of a fitness measure of the planning process would be the timeliness of the plan, that is, when the

[30] Field Marshal Helmuth von Moltke

[31] Hayes, Richard E., Richard L. Layton, William A. Ross, Jan W.S. Spoor, and Theresa A. Hollis. *Enhancements to the Army Command and Control System*. Vienna, VA: Evidence Based Research, Inc. 1993.

plan was produced relative to the need for a plan given the situation. Two fitness measures for plan quality are its feasibility, the extent to which the plan can be implemented and its relevance, the extent to which the plan relates to achieved intent in the context of the situation.
In general, it is easier to collect data that provides values for objective measures, like the time it takes to produce a plan, and more difficult to assess fitness measures, such as relevance or feasibility. The exception is when data is being generated by simulations or in controlled experiments. In these cases, when ground truth is known, it is far less difficult to determine values for fitness measures. A conceptual model and an instrumented working model that instantiates the appropriate variables and relationships provide what is required to generate the data that is needed.

Agility

Agility is the appropriate response to uncertainty and risk. Agility is an attribute of a process and the products of a process. Agile command and control, sensemaking, planning, and execution individually and in combination guard against surprises, are more likely to be responsive, can make up for not getting it absolutely right initially, and thus, other things being approximately equal, will tend to be both more effective and efficient. Agility is a scenario-independent measure of merit, one that should figure prominently in any campaign of experimentation designed to explore approaches to planning and plans.

Agility is a multidimensional concept that is predicated on the achievement of a threshold level of effectiveness. In other words, one cannot be ineffective *and* agile. The dimensions of agility[32] include: responsiveness, robustness, flexibility, innovativeness, resilience, and adaptability.

A Campaign to Explore Planning for Network-Centric, Effects-Based Operations

The goals of the campaign outlined here are to (1) understand the nature of a planning approach that is appropriate for effects-based, Network Centric Operations, and (2) assess the relative merits of various approaches to planning in this context.

The operations considered in the campaign should involve complex civil-military missions undertaken by a broad coalition including nation states, international organizations, and various non-governmental and private voluntary organizations (NGOs and PVOs).

This campaign will consist of four phases: a Formulation Phase, a Concept Phase, a Refinement Phase, and a Demonstration Phase, and involve interrelated research, analysis, and experimentation activities. These activities will be centered about a conceptual model that places planning in the context of Network Centric Operations and the quality of a plan in the context of the network-centric value chain.

---

[32] Alberts and Hayes, *Power to the Edge*. Chapter 8.

Formulation Phase

Although this document provides a vision for a campaign of experimentation focused on exploring planning for network-centric operations and an initial statement of a conceptual model that can be used to guide the campaign and design individual experiments and analyses, the formulation of this campaign cannot be completed until a decision is made to undertake it. Following such a decision several steps need to be immediately taken. These include selecting the identity of the set of participants, establishing their roles and responsibilities, and allocating the resources that will be devoted to the campaign.

In anticipation of a decision to proceed, this document provides the outline of a plan of research, analysis, and experimentation activities needed to achieve the objectives of this campaign.

Concept Phase

A properly conceived and executed Concept Phase helps ensure the success of the campaign. The following tasks need to be undertaken during the Concept Phase:

- Developing a conceptual model
- Establishing the baseline
- Identifying and characterizing promising approaches to planning
- Developing approaches and instruments for key variables

Of these, the first two should be given top priority. Their completion is absolutely essential prior to the conclusion of the Concept Phase.

Developing a Conceptual Model

Key components of a suitable conceptual model have been presented earlier in this document, but the components need to be integrated and fleshed out before the model is suitable to form the intellectual core of this campaign of experimentation. The objective is to construct a conceptual model that contains all of the variables that are expected to have a first order effect on the effectiveness of an approach to planning. The model also needs to incorporate a value chain that is capable of distinguishing between network-centric and traditional approaches to planning. In addition, care must be taken to ensure that the model contains the variables necessary to characterize the range of missions/operations of interest. Finally, the planning approach space needs to be mapped to specific ranges of values for a set of variables that capture the dimensionality of the planning space and specific approaches starting with the baseline, the current ATO process, and an edge approach to anchor the opposite corners of the space.
A NATO research group (SAS-050) has spent the last 2 years developing a conceptual model[33] that can be used to explore and assess new approaches to command and control. This group has identified over 300 variables and the significant relationships among these variables. They have also created a value view that identifies the links in a value chain that begins with the capabilities

[33] "The Conceptual Model of Command and Control." Prepared by NATO's Research and Technology Organization's Studies Analyses and Simulation Panel (RTO SAS-050). June 2003 to November 2005.

and characteristics of a force. The purpose of this reference model is to serve as a checklist for those constructing a conceptual model such as the one required for this campaign of experimentation. Clearly, not all of the over 300 variables and associated relationships will be directly applicable, but to a high degree of probability, virtually all of the variables that are relevant to this effort can be found in this reference model. The task at hand is to select a subset of these variables and relationships to serve as an initial conceptual model for this effort.

Establishing the Baseline

The second critical task that needs to be completed during the conceptual phase is the establishment of a quantitative baseline. Starting with the subset of the planning space that reflects the current approach to ATO planning, the values of the variables that comprise the value chain need to be estimated and the nature of key relationships (e.g., how the current planning process is affected by the quality of information or the need to respond to a dynamic situation, that is, produce or change a plan in x hours). This involves reviewing, analyzing, and most likely gathering empirical evidence and/or modifying and running available simulation models.[34] This campaign baselining activity is in addition to having an explicit baseline to anchor both the individual analyses and experiments that will be undertaken.

Because appropriate data and/or models may not be readily available, baselining may need to involve the development of new simulation models and undertaking a set of specifically designed experiments and analyses. It should be noted here that objective quantitative data is critical to the success of the campaign. Developing and moving to a network-centric approach to planning should not be undertaken because selected subject matter experts (SMEs) feel good. There are no SMEs for Network Centric Operations simply because we do not yet have enough experience with these approaches.

Identifying Promising Approaches to Planning

It is not reasonable to expect that all of the promising approaches to planning will be identified during this phase. However, at least one planning approach should be identified in four distinct regions of the planning space:

- The region immediate adjacent to the baseline or current approach,
- The region that contains edge approaches to planning that have the characteristics of Level 4 of the network-centric maturity model,
- A region that represents a capability that corresponds to Level 2 of the maturity model, and
- A region that corresponds to Level 3.

The first two bound the problem, while the last two provide at least an initial idea of the shape of the fitness curve.

[34] It is far more likely that a simulation that reflects existing processes and approaches will be available than one that is able to reflect new approaches or can easily be modified to accomplish this.

It should be expected that some of the approaches identified will, in all likelihood, be discarded or modified, or replaced by others that incorporate what seems to work and changed by what does not seem to work. One should not lose sight of the fact that changes to the approaches under consideration are, in fact, signs of progress and not signs of failure.

Developing Measurement Approaches and Instruments

Prior to the conduct of experiments, it is necessary to have developed the approaches and instruments necessary to collect data about the variables of interest. These are required in order to establish the values of these variables under certain conditions. Objective data regarding key concepts and variables associated with the network-centric value chain have only recently been collected, and thus mature and validated approaches and instruments do not yet exist. Measuring the right variables and understanding the characteristics of these measures is a critical component of experiments. The difficulty of this task should not be underestimated and it should be expected that multiple attempts will be needed to arrive at a satisfactory approach to a particular variable or set of variables.

Quality metrics for awareness, shared awareness, planning, a plan, and synchronization each present a somewhat different challenge. Trying to measure awareness involves taking a measure in the cognitive domain, a domain that is not directly accessible. Thus, instruments used to measure awareness are designed to elicit responses from subjects and/or observe their behavior. This means that the measurements will be indirect ones from which awareness is inferred. Shared awareness has all of the challenges associated with measuring awareness as well as the problem of operationally defining what is meant by *shared*. Planning, an activity or process, can be measured by the degree to which the process specified is carried out in reality, but this, in and of itself, is not a sufficient measure of the quality of a planning process. This is a measure of the quality of the execution of a specific planning process, not the quality of the planning process. It is the latter measure that is needed in order to be able to compare different approaches to planning. The quality of a plan (the expression of the result of a planning process) requires both types of measures, that is, a measure of how good a “type x” plan it is, as well as how good a plan it is given the situation.

Synchronization is a concept that can be applied to decisions, actions, or effects. While it may be relatively easy to define the endpoints of a scale, the difficulty lies in defining how far from the endpoint a given instance is. In the case of synchronization, the scale to be used should range from a large negative (corresponding to conflicts among, for example, the decisions made) to a large positive (corresponding to synergies achieved). The midpoint is a zero, a point that represents deconfliction (where there are no conflicts, but also where there are no synergies).

Refinement Phase

This phase of the campaign of experimentation tests, assesses, and refines the planning concepts under consideration. In the associated program of research, efforts are focused on increasing the understanding of key relationships (e.g., the set of circumstances, the quality of planning, and the quality of the plan produced for different approaches to planning).

The object of this phase is to develop a solid empirical basis for proceeding with the implementation of a set of planning approaches[35] and their integration into existing organizations. Establishing a solid empirical basis for moving ahead with a particular approach involves (1) exploring the various options over a wide set of circumstances that reflect the range of missions that DoD can be expected to undertake, (2) replicating experiments to develop sufficient experience and data, and (3) maturing the various approaches so that they reflect their potential. As was stated previously, this involves pushing ideas and capabilities to their breaking point and developing an understanding of why various approaches or concepts (treatments in the language of experimentation) either work or do not work. Establishing a solid empirical basis also requires that experimentation activities be conducted in accordance with the applicable Codes of Best Practice.

The products of this phase include (1) a refined and enriched conceptual model and (2) capabilities that are ready to demonstrate.

Demonstration Phase

This phase provides the opportunity to involve a variety of individuals and organizations, either as observers or participants, to make them aware of and to understand the new concepts and approaches that merit implementation. While there will be opportunities to have selected individuals and organizations observe and even participate in the earlier phases, this is the time to expose what has been developed and learned during the course of the campaign. It may seem obvious, but demonstrations have little value if the new approach to be demonstrated has not been adequately tested, if the supporting systems are not stable, and if the participants (those involved in the demonstration) have not been adequately trained. Demonstrations also have little or no value if the problem or situation is insufficiently complex or interesting and if it cannot be objectively shown that the new approach is better. In this case, it would be preferable to run side by side demonstrations of how groups of individuals could, for the same situation, plan for and execute a variety of missions.

Priorities for a Program of Research

Given that there are many things about network-centric approaches to operations and planning for these operations that are not well understood, a program of focused research must necessarily accompany this campaign of experimentation. While the campaign of experimentation will provide empirical evidence regarding the desirability of alternative approaches to planning for NCO, the associated program of research will explore basic relationships among key variables, how best to measure them, and incorporate what is learned into the conceptual model.

[35] This refers to a set of planning approaches because it is not envisioned that one approach will be best suited for all circumstances.

Plan of Work: Concept Phase

This section identifies and discusses the tasks that need to be accomplished during the Concept Phase and their relationship to one another (see Figure 13). It assumes that the campaign has been formulated, that the team has been assembled, and that peer review and advisory groups have also been assembled.

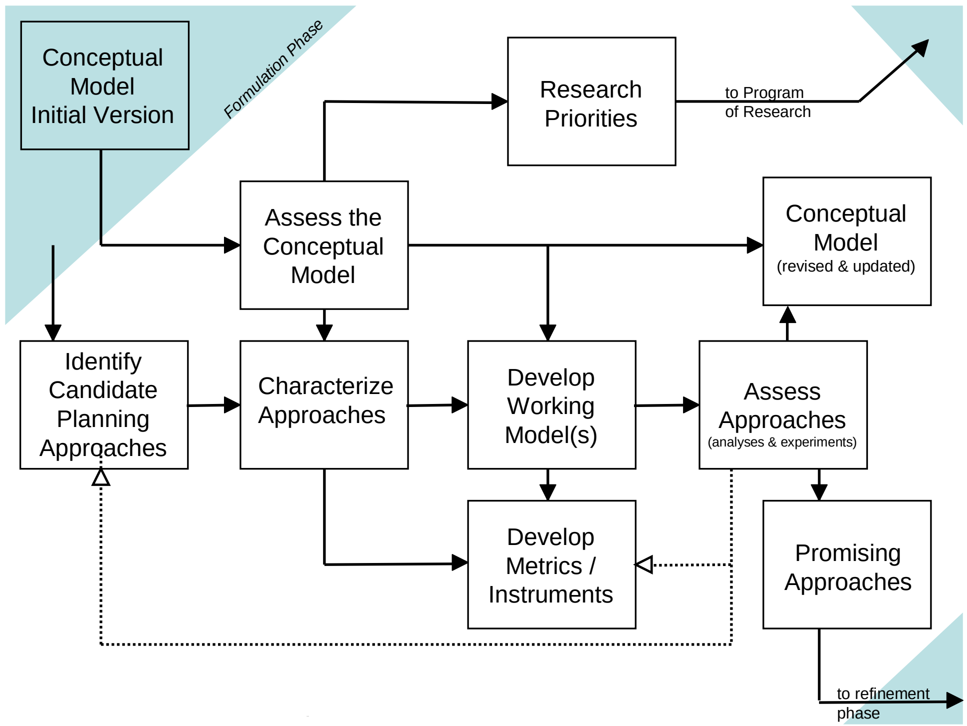


Figure 13: Plan of Work: Concept Phase

Assessing the Conceptual Model

The conceptual model plays a central role in the campaign. Its state at the beginning of the campaign provides the point of departure. In large part, it determines what the priorities are for analysis, experimentation, and the program of research. As the model improves, it becomes ever more useful in analyses that extend specific empirical results and provide the basis for sensitivity analyses of the assumptions made in the experiments.

The first step in the research program is the determination of research priorities. These priorities are a product of an assessment of the conceptual model. The primary focus of the program of research needs to be on the relationships that are not well understood, particularly those that involve the variables that define the characteristics that distinguish one approach to planning from another, and those related to the value chain.

Identifying and Characterizing Candidate Planning Approaches

The planning space depicted in Figure 6 should be used to identify and characterize approaches to planning that will initially be considered during the Concept Phase. While the planning space serves to provide information about the three key dimensions of a planning approach (the nature of the planning process, nature of the plan, and information dissemination and sharing), fully specifying a planning approach involves actually developing the details of the process and the plan produced. It also involves describing those elements of a mission capability package that are directly related to planning (e.g., command approach: how intent is developed and communicated) as well as those that use the products of the planning process. The latter are needed to be able to understand and assess the relationships between planning and execution.
As discussed earlier, it is best practice to consider, at least initially, approaches from all regions of the planning space. One way to ensure that the choices are not overly constrained by current views is to make sure that at least one approach representing each level of network-centric maturity is identified and considered.

Discovery experiments[36] are often used to instantiate workable versions of planning approaches, particularly those that have never been used before. Discovery experiments informed by lessons learned can be used to improve upon existing approaches or modify them so that they evolve into a part of the planning space in the vicinity of traditional approaches, but are a step toward a network-centric approach.

Developing a Working Model

The conceptual model components provided here and the conceptual models that were developed by NATO RTO and OSD identify key variables that are required to understand and test a variety of approaches to command and control. They were each developed with the objective of being able to compare traditional and network-centric or edge approaches. They provide a checklist of variables and relationships to be considered. As such, these conceptual models are not in a form that is suitable for direct application in specific analyses and experiments. First, for many efforts they contain a great many variables that are either of little or no interest. Second, these are not executable models, that is, they are not in a form that allows one to set the values of certain variables and see what effect they have on other variables. For these reasons, working models based on these conceptual models need to be developed for analyses and experiments.
A working model contains, as indicated above, a subset of variables and is in a form that facilitates manipulation. Different types of models (simulations, agent-based, spreadsheets, system dynamics) have different purposes and, most often, several are employed in the same effort.

---

[36] Alberts and Hayes, *Campaigns of Experimentation*. p73.

Developing Metrics and Measurement Instruments

Concepts such as information quality have been an integral part of analyses and experiments for a considerable time. Despite this, a number of advances in our understanding of what constitutes information quality have taken place recently. For example, it has been recognized that terms like *currency* and *timeliness* were being used rather loosely and interchangeably, despite the fact that they have different meanings. The difference between these two terms is that one is an objective measure of the passage of time while the other makes sense only in the context of a specific situation and thus is a fitness measure. This distinction is not confined to these two measures. A re-examination of how we think about various quality metrics has led to the separation of objective and fitness measures. As we became more sensitized to the issue of information assurance, there was a growing recognition of the need to add IA-related attributes. Figure 14 provides an up-to-date view of the attributes of information quality.

- Objective
  - Correctness
  - Completeness
  - Precision or accuracy
  - Currency
  - Consistency
- Information Assurance
  - Privacy
  - Integrity
  - Authenticity
  - Availability
  - Non-repudiation
- Fitness for Use
  - Relevance
  - Timeliness

Figure 14: Attributes of Information Quality

The task here is to review the definitions of all of the variables selected for the working models and, if any are not deemed satisfactory, pursue a research effort to improve upon the existing definition, and if they are satisfactory, review available instruments and modify or improve upon these as required. At a minimum, the variables that should receive priority attention include those included in the figures in the initial conceptual model for network-centric planning presented here.

Assessing Candidate Approaches

At this point in the campaign, there will be a fairly large number of candidate approaches. It is better to err on the side of too many rather than too few. The approach to assessment that makes sense in this situation is to use a variety of methods and tools designed to quickly explore the potential value of the alternatives. These include a variety of model-based analyses and well-instrumented discovery experiments. The current practice of using poorly instrumented and poorly structured games with subject matter experts is problematical. A major danger is that "familiar and comfortable" will substitute for potential value as the operable selection criteria.

It is important when conducting these initial assessments that the candidate approach has an opportunity to mature; the bugs need to be worked out. Not allowing ample time and allocating sufficient resources for maturation will bias the results. Similarly, time is needed for subjects to familiarize themselves with the new approaches.

It is also important to use, for each candidate approach, a variety of mission challenges. This is necessary to make sure that an approach has promise over a reasonable portion of the mission space.

The cost of these assessments is a function of the number of individuals involved and the number or runs. The more analysts, experimenters, and subjects involved, the greater the costs will be, and with increased costs comes a reluctance to replicate the activity. This is why model-based analyses should both precede experiments that involve many subjects and also follow them in order to replicate and extend the results of these experiments.

Plan of Work: Refinement Phase

As a result of efforts in the Concept Phase, a number of promising candidates will have been identified. In this phase, each of these approaches will receive a more indepth look. This involves the conduct of a series of coordinated analysis-model-experiment-model-analysis efforts, one for each of the promising approaches. In additional to the efforts focused on specific approaches, a concurrent analysis effort is needed to synthesize the results obtained and to ensure that valid and interesting comparisons can be made between and among the most promising of the approaches. As a result of these efforts, it can be anticipated that areas and issues that require research will be identified. This synthesis effort, as well as the indepth looks at promising approaches, should be informed by the results of research. Figure 15 depicts the activities of the Refinement Phase.

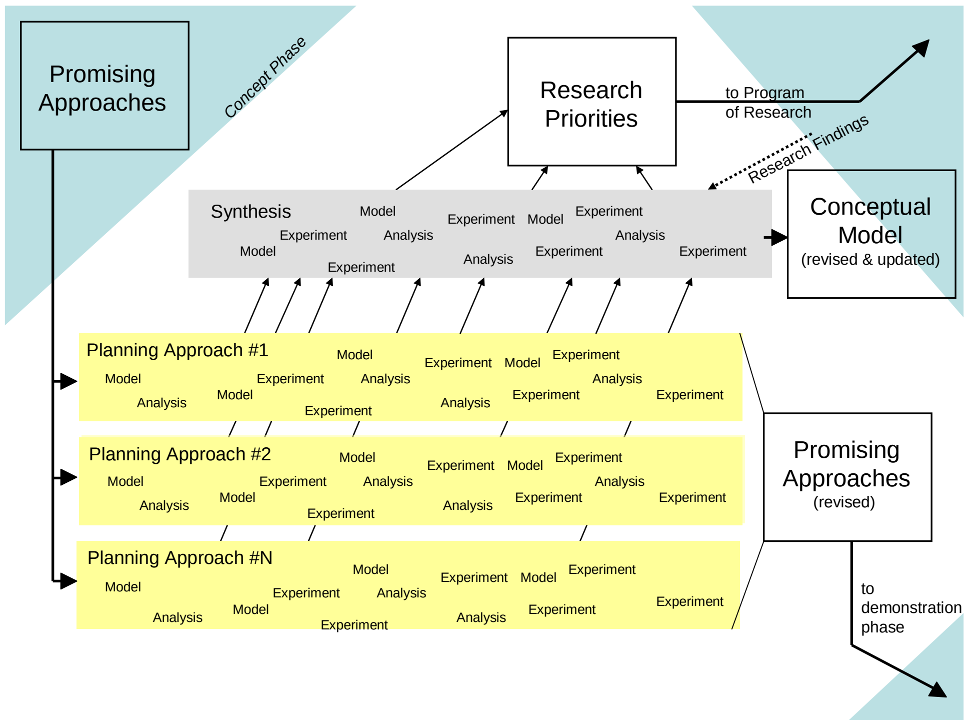


Figure 15

In-depth Investigations

Successful indepth investigations (analysis-model-experiment-model-analysis) are not only those that find a promising approach to be truly promising, but also include those that result in learning why an approach may not work, or work only under very specific circumstances. In fact, an indepth investigation that simply confirms the promise of an approach and does not result in an understanding of why it works and under what circumstances it does not work (the boundary conditions) is *not* a success, despite the fact that the approach continues to show promise. The "model" referred to here is a working model derived from the conceptual model. The analyses conducted as part of the indepth investigation include those that (1) explore the approach and conditions space to identify areas that should be the subject of experiments and (2) are devoted to extending the results of the experiments by performing sensitivity analyses.

Synthesis

The synthesis activity is important for four reasons.

- First, to help ensure that the lessons learned are communicated across approach investigations.
- Second, to be able to systematically capture the results and findings of these investigations and incorporate them into the analysis of planning in Network Centric Operations.
- Third, to serve as a basis for assessing the progress being made and guidance regarding directions to these investigation, and
- Fourth, to incorporate the data collected and the results of analyses into an updated conceptual model and associated knowledge base.

Program of Research

There is always a tension between a desire to produce immediate results, that is, improve current capabilities in near-real-time and mid- to longer-term objectives. It should be remembered that we need disruptive innovation to meet the challenges of the 21st century and the more disruptive a new approach is, the more time will be required to mature the approach, and then assess and field it. Thus, the approaches we can spin off early on are more likely to be incremental improvements and if too much time and too many resources are devoted to satisfying these short-term demands, it will adversely affect our ability to deliver the disruptive innovation we require. The program of research is necessary to lay the foundation for disruptive innovation and enable us to, as time goes by, be able to generate approaches to planning more rapidly with less risk. As such, this program of research should begin as soon as possible, focusing on the gaps in our understanding that we currently have.

Determining Research Priorities

In due time, the campaign-specific priorities for the program of research will emerge from an assessment of the conceptual model, the experience associated with developing working models, and their capability to inform efforts to assess specific approaches to planning.
However, NCO has been under consideration for some years now and we have a reasonable appreciation of the major issues and uncertainties that we face. The OSD and NATO efforts to build a conceptual model that facilitates the exploration of network-centric concepts and capabilities and comparisons with traditional approaches to command and control have resulted in a solid foundation upon which to build. These models identify key variables and relationships needed to understand the role that planning, as an element of sensemaking, and plans, as products of command and control, play in Network Centric Operations.

Clearly, anything that would contribute to a more complete, useful, and believable conceptual model is potentially of considerable value because it would (1) help focus initial experiments and analyses, and (2) allow model-based analysis to play a larger role and lessen the need to rely on costly experiments.

High Priority Research Topics

It is important that we do not confuse our current notions of planning and plans that have coevolved with current organizations, processes, and products with what is possible and what may be best suited to NCO. Therefore, some basic research is in order to identify and systematically explore the possibilities.

Research topics that should be given high priority include:

- Taxonomy for Planning and Plans
- Quality Metrics for Planning and Plans
- Factors that Influence Planning Quality
- Factors the Influence Plan Quality
- Impact of Planning and Plan Quality on Operations

- Methods and Tools for Planning
- Plan Visualization

Taxonomy for Planning and Plans

Some work has been done on characterizing various approaches to command and control and mapping specific approaches taken by different countries at different times to a C2 approach. Although centralized planning is identified as a feature of Industrial Age approaches to command and control[37] and coupled with decentralized execution, there has not been any comprehensive treatment of planning as a function of C2 approach.

To facilitate the exploration of planning and plans in operations, specifically Network Centric Operations, these taxonomies (one for planning and one for plans) are required.

These taxonomies need to focus on what makes various approaches to planning and different kinds of plans different. Once these two taxonomies are completed, it is important to test these taxonomies by seeing if they are able to map past, current, and proposed approaches to specific classes of planning and plans defined by the taxonomies.

Quality Metrics for Planning and Plans

It is hard to overestimate the importance of having appropriate metrics. The quality of information is, for example, not an adequate substitute for the quality of awareness. Expert opinion, particularly when the expertise may not be on point, is not, in and of itself, an adequate substitute for performance or value. Attention to metrics early in the campaign will greatly enhance the value of the analyses and experiments that are undertaken.

While it is important to define metrics that measure what needs to be measured, there will be cases when these metrics are either difficult or costly or both to measure. The correct response is find indicants that can serve as an approximation and study the relationship between these indicants and the metric of interest. The incorrect response is to ignore the metrics that present measurement problems. It is far better to get an approximation for the correct metric than to get a precision measure of the wrong metric.

Factors that Influence Planning Quality

There are, of course, many factors that influence the quality of planning. The objective of this research is to sort out those that have a first order effect. The DoD Network Centric Conceptual Framework and the NATO SAS050 Conceptual Reference Model provide points of departure, presenting lists of candidate variables and relationships. In looking at these, it should be noted that these models are focused on decisionmaking and decisions rather than planning and plans. Thus, an interesting topic for investigation might be the differences, if any, between the subset of

[37] Alberts and Hayes, *Power to the Edge*. pp46-47.

decisions that are related to planning and expressed in plans and decisions that are not directly related to planning.

It is important to remember that planning is a cognitive and social process and thus it is the interactions among people and organizations are of utmost importance. Analyses that focus on information quality and flows are unlikely to result in an adequate understanding of planning processes, although these factors clearly need to be included.

Factors that Influence Plan Quality

Although the quality of the planning process can expected to be a major factor if not *the* major factor in determining plan quality, other factors need to be considered. Although, in theory, the approach to command and control and command arrangements includes the approach to planning and the nature of any plans that are produced, in practice, inconsistencies can occur, particularly in coalition and civil-military operations. Thus, attention also needs to be paid to the compatibility between the nature of the plan (e.g., level of detail), command arrangements, and the approach being taken to command and control.

Impact of Planning and Plan Quality on Operations

The relationship among the quality of planning, plan quality, and operations is complicated by exogenous variables that have a direct impact on operations. Thus the investigation into the impact of the quality of planning and the quality of plans needs to focus on determining the conditions under which the quality of planning/plans has a dominant influence over operations and those where the influence of planning/plan quality is muted.
It is likely that the relationship between planning/plan quality and operations differs as a function of the approach to command and control. Therefore, the approach to command and control needs to be a controllable variable in these investigations.

Methods and Tools for Planning

The complexity of operations has and will continue to increase while, at the same time, windows for effective responses have and will continue to shrink. This has put enormous pressure on planning processes to produce plans that adequately deal with the complexity of the situation in less and less time. Methods and tools have the potential to help planners deal with both these increases in complexity and the requirement for more timely plans.
One of the sources of increased complexity is the set of differences that exists between planning processes and the information systems and flows that support planning. Methods and tools designed to bridge these differences and minimize their adverse effects should be one of the focuses of this research.

Given the consequences of a failure to deconflict plans or actions, it is not surprising that a lot of attention has been focused on processes and methods that are designed to ensure that conflicts and mistakes do not occur. However, we cannot be satisfied with a planning process that does not move beyond deconfliction to realizing synergies. Research on new methods and tools is

needed to see if we can improve synergies while providing adequate protection against conflicts and mistakes, while at the same time reducing the time it takes to develop plans.
Particular methods and tools may be more suitable for some approaches to planning and/or types of plans than others. Therefore, as part of this research, it is important that we understand when specific methods and tools are useful and when they are not.

Plan Visualization

A key factor determining the success of a plan is likely to be the ability of participants in the operation to understand the plan and developing shared awareness. Hence, anything that helps people to visualize a plan and fully understand its implications and what it means for them should be of value.

It should be noted that one size does not fit all when it comes to visualization approaches or visualizations. Thus, an important part of this research will be what visualization techniques are useful as a function of different situations, roles and responsibilities and individual characteristics (including experience).

Part II: The Way Ahead: Critical Path

Overview

Developing a better understanding of the class of command and control arrangements and planning processes that work well with network-centric and coalition operations is on the critical path to DoD Transformation. To improve our current understanding, we must improve (1) our models, (2) our ability to measure key variables, and (3) our ability to conduct analyses and experiments. In addition, we also need to simultaneously push both the state of the art and the state of the practice of command and control itself. The state of the art is generally understood to be on the critical path, but the connection between understanding C2 and improving the practice of C2 is not as widely recognized. The campaign of experimentation and associated program of research are designed to contribute to all of these objectives.

Synergies: Key to Accelerating Progress

The experimentation campaign and associated program of research should not be undertaken in isolation from the practice or from practitioners. Figure 16 depicts the synergies that result from an effort that involves researchers, experimenters, and practitioners.

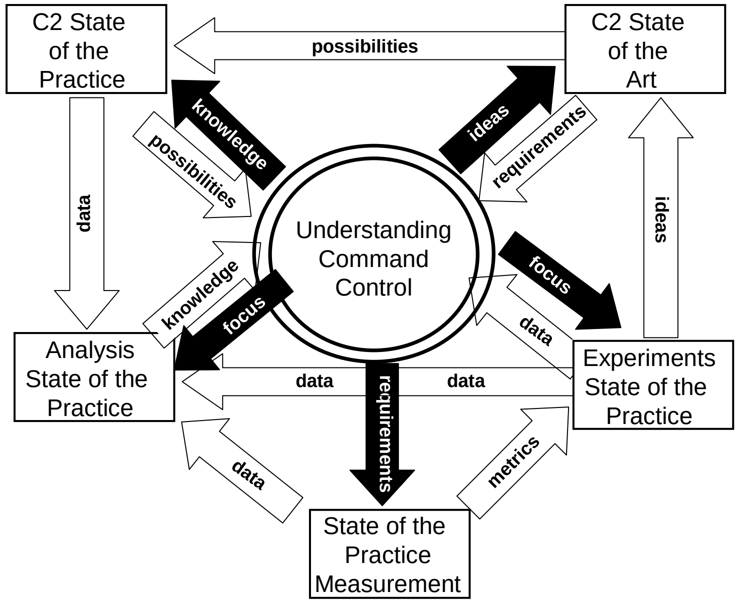


Figure 16: Synergies in a Multi-Pronged Effort

Improving the state of the art of planning (the theory) serves as a source of possibilities for practitioners (applications of theory). At the same time, developments in the field that improve practice also inform theory and improve our understanding of network-centric planning. Improving the practice not only can generate a set of possibilities that theory needs to address, but also provides opportunities to collect data that can inform our models and give us opportunities to test instrumentation and measurement approaches. Improvements in understanding provide knowledge to practitioners, ideas that can improve the state of the art, a focus for experiments and analyses, as well as variables and relationships that need to be measured. Improvements in the analyses we undertake can provide knowledge that enables us to improve our understanding while improved experimentation provides valuable data that can enhance our understanding. Finally, improving our ability to measure key variables and relationships can help experimenters by giving them the tools they need and help analysts by providing better data. A failure to invest and harvest the advances made in any one of these areas diminishes our ability to undertake these activities, makes them less productive, and retards overall progress. Thus, the way ahead involves investments in each of the areas depicted in Figure 16 and the creation of a mechanism for sharing possibilities, data, ideas, knowledge, metrics, measurement techniques and tools, and, most significantly, understanding.

Establishing a partnership between those who design and conduct campaigns of experimentation and those who undertake focused programs of research and practitioners is necessary if these synergies are, in fact, to be realized. Thus, when crafting a plan for a campaign of experimentation and an accompanying program of research, there must be a section of the plan devoted to engaging and involving practitioners at the beginning of the process rather than waiting, as so often is the case, until the phase of the campaign when demonstrations are conducted. Practitioners need to be offered an opportunity to contribute to the exploration of various approaches to planning (the planning approach space) and not just shown the “solution.”

The Critical Path

In recent years, interest in the practice of C2, developing a theoretical understanding of C2, and exploring new approaches to C2 have increased significantly. This, in large part, is due to a growing appreciation that DoD Transformation is, in fact, an Information Age transformation and that the adage that new ways of doing business are required means that new approaches to C2 are required. New approaches are required to adapt existing mindsets and practices to the security challenges of the 21st century as well as to fully support emerging concepts of operations (e.g., network-centric and effects-based).

As a result, individuals and organizations around the world are engaged in a wide variety of research and analysis activities. They are producing useful data and creating bits of knowledge. It is vitally important that these C2-related research and analysis activities not only be supported but expanded as well. Equally, if not more important, is the urgent attention that needs to be paid to creating the conditions necessary to get the most out of the C2-related research and analyses that are undertaken. Only by making the most of the investments we make can be hope to accelerate progress.

Research, analysis, and experimental efforts are both more efficient and more effective if they can build upon data that has already been collected and knowledge that has already been created. Clearly taking full advantage of available data, research and analysis findings, and the existing body of knowledge requires that individuals and organizations become aware of what is available. Progress depends then on the level of shared awareness in the C2 community. However, simply knowing that some data was collected or that some analysis or experiment was done, does not, in and of itself, make these data and findings useful. The utility of the data collected depends upon the following:

- The existence of metadata
- The relevance of the metrics used
- The appropriateness of the instruments
- The conditions under which the data was collected

The value of analyses and experiments depend upon the quality of their formulation and the extent to which the efforts adhered to the Code of Best Practice for C2 Assessment. Similarly, the value of the experiments that have been done depends on how well they were designed and conducted. Both of these, in turn, are a function of the degree of adherence to the Code of Best Practice for Experimentation. The value of both the analyses and experiments that are conducted also depend on the quality of the conceptual and working models that are employed. These in turn depend on the quality of an evolving C2 Conceptual Reference Model, which is, in the final analysis, a community effort. Thus, the prevailing state of the practice of analysis and experimentation determines the rate of progress as much as the degree to which the community has shared awareness of what exists, what is ongoing, and what is planned.

Priorities

Understanding network-centric planning requires a community effort. It requires increased collaboration and cooperation between and among individuals and organizations that are

interested in defense transformation in general and specifically, those interested in new approaches to planning (as part of a C2 approach) that anchor coevolved network-centric mission capability packages. Three areas warrant immediate, priority, and sustained attention:

- The initial conceptual model
- Adopting the Codes of Best Practice
- Establishing a portal for data, findings, and instrumentation

Initial Conceptual Model

There have been two major efforts to develop a conceptual model that can be used to organize existing knowledge, focus research and experimentation, and support analyses related to an Information Age transformation. The first, sponsored by OSD (a collaboration of the Office of Force Transformation and the Command and Control Research Program in the Office of the Assistant Secretary of Defense for Networks and Information Integration) used the tenets of Network Centric Warfare as a point of departure for constructing a conceptual framework that could be used to structure case studies and convey what transpired in a systematic manner that facilities comparisons between traditional and network-centric approaches to operations. The second effort, under the sponsorship of the NATO Research and Technology Organization's Studies, Analysis, and Simulation Panel (research group SAS-050), independently[38] developed a C2 conceptual model designed to facilitate the exploration of new, network-centric approaches to C2. This initial version of the NATO C2 Reference Model was then validated by (1) applying the NATP COBP for C2 Assessment model to a case study to assess its utility (ability to support problem formulation), (2) an extensive literature search conducted to identify variables that were found to be relevant to C2 and its relationship to operations, and (3) comparing the NATO model to the OSD Conceptual Framework. As a result, the current version of the NATO C2 Reference Model represents the best thinking of a set of international experts[39] and provides the community with a conceptual model to employ in research, analyses, and experiments and a firm foundation to build upon.

The NATO C2 Conceptual Reference Model is accessible on the CCRP Web site (www.dodccrp.org) and this Conceptual Reference Model should be used as a source of ideas and a checklist to help ensure that this campaign of experimentation and its associated program of research consider all of the variables and relationships that are relevant to their efforts.

Adopt Codes of Best Practice

Many years of effort have been devoted to the development of the three Codes of Best Practices currently available (COBPs for C2 Assessment, Experimentation, and Campaigns of Experimentation). Their value is clearly a function of the extent to which individuals and organizations are committed to adopting and adhering to them. These codes should be distributed

[38] These efforts were largely independent, although key members of the NATO group including its chairman participated in both efforts. However, the Chairman insisted that the NATO group start with a clean sheet of paper and build their model from the experience of the participating analysts who came from both NATO and non-NATO countries.

[39] Experts from the United States, Canada, Sweden, France, Germany, Denmark, the Netherlands, Norway, Spain, Turkey, the United Kingdom, and Austria.

to all participants and classes formed to familiarize individuals and organizations with their content and application (SAS-050 members have conducted such courses and arrangements can be made through the CCRP for customized versions of these courses).

Establish Portal for Data, Findings, and Instrumentation

Power to the Edge principles[40] that are now embodied in DoD Policy and Directives, include the concept and practice of moving from smart push to smart pull. This shift in the approach to information dissemination is designed to promote widespread information sharing and collaboration, a necessary condition for attaining shared awareness. A cornerstone of this shift in responsibilities is the requirement for individuals and organizations, in this case, the researchers, analysts, and experimenters engaged in this effort, to post in parallel. Of course, this is insufficient to achieve the objective of providing users with the opportunity to shape their own information positions. This is because the information not only has to be available for users to access, but users also need to know what information is available and where and how to get it.[41] One way to accomplish this is the creation of a portal with an accompanying effort to make its existence widely known. This portal should provide access and links to data, findings, and instruments of interest to researchers, analysts, and experimenters. The CCRP is working to put together a community portal designed to help in this regard.

Final Thoughts

The undertaking of a campaign of experimentation as well as a related program of research is a significant effort. For this to be successful, adequate resources must be provided and it must be allowed to proceed paced by the results and findings it develops and not by some arbitrary schedule. Learning how to plan in the context of Network Centric Operations (particularly coalition operations) is decidedly not about how planning has been accomplished in the past. Rather it is about what functions planning needs to accomplish, the potentially useful approaches to planning, and the value propositions that trace improvements in planning to improvements in C2 to measures of operational effectiveness, particularly agility.

[40] Alberts and Hayes, *Power to the Edge*. p82.
[41] And of course, the information is not of value unless metadata is also provided.